

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**  
**КАФЕДРА ОРГАНІЗАЦІЇ АВІАЦІЙНИХ ПЕРЕВЕЗЕНЬ**

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач кафедри

Г.М.Юн  
“ \_\_\_\_\_ ” \_\_\_\_\_ 2020 р.

# **ДИПЛОМНА РОБОТА**

**(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ  
«МАГІСТР»**

**Тема: Організація доставки термінових вантажів за допомогою  
безпілотної авіації**

Виконавець: Петрук Владислав Олександрович

Керівник: Юн Геннадій Миколайович

Консультанти з окремих розділів пояснювальної записки:

Аналітична частина Юн Г.М.

Проектна частина Юн Г.М.

Нормоконтролер: Жукова Світлана Олександрівна

**Київ 2020**

**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**Факультет Транспорту, менеджменту і логістикиКафедра Організації авіаційних перевезеньСпеціальність 275 «Транспортні технології (на повітряному транспорті)»ОПП: «Організація перевезень і управління на транспорті (повітряному)»  
(шифр, найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Г.М. Юн

«\_\_\_\_\_» \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ****на виконання дипломної роботи****Петрука Владислава Олександровича**

(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема дипломної роботи «Організація доставки термінових вантажів за допомогою безпілотної авіації» затверджена наказом ректора від «17» жовтня 2019 р. № 2401/ст

2. Термін виконання: з 14.10.2019 по 29.12.2019р., з 20.01.2020 по 09.02.2020 р.

3. Вихідні дані до проекту: статистичні дані щодо БПЛА, дані про перевезення вантажів за допомогою БПЛА, класифікація БПЛА, термінові вантажі.

4. Зміст пояснювальної записки: статистична інформація щодо розвитку БПЛА у перевезеннях термінових вантажів, побудова моделі перевезення термінових вантажів за допомогою БПЛА.

5. Перелік обов'язкового графічного (ілюстративного) матеріалу: графічне представлення БПЛА. Графічне представлення та порівняння статистичних даних наведених у аналітичній частині.

## 6. Календарний план-графік

№ пор.	Завдання	Термін Виконання	Відмітка про виконання
1.	Збір та обробка статистичної інформації	14.10.2019	10.11.2019
2.	Написання теоретичної частини	11.11.2019	18.11.2019
3.	Написання аналітичної частини	19.11.2019	02.12.2019
4.	Написання проектної частини	03.12.2019	15.12.2019
5.	Написання вступу та висновків	16.12.2019	21.12.2019
6.	Оформлення пояснювальної записки	22.12.2019	03.01.2020
7.	Оформлення графічного матеріалу та презентації	04.01.2020	18.01.2020

## 7. Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Теоретична частина	Юн Г.М.	14.10.2019	14.10.2020
Аналітична частина	Юн Г.М.	19.11.2019	19.11.2019
Проектна частина	Юн Г.М.	03.12.2019	03.12.2019

8. Дата видачі завдання: « 14 » 10 2019 р

Керівник дипломного проекту \_\_\_\_\_ Юн Г.М.

Завдання прийняв до виконання \_\_\_\_\_ Петрук В.О.

## РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Організація доставки термінових вантажів за допомогою БПЛА» складається з 134 сторінок, містить 34 рисунки, 9 таблиць, 46 літературних джерел.

**ПІДПРИЄМСТВО, БПЛА, ВАНТАЖІ, МОДЕЛЬ ДОСТАВКИ ТЕРМІНОВИХ ВАНТАЖІВ ЗА ДОПОМОГОЮ БПЛА.**

*Метою роботи є організація доставки термінових вантажів за допомогою безпілотної авіації. Для досягнення поставленої мети у роботі необхідно виконати низку завдань:*

- описати сутність та структуру БПЛА;
- дослідити організацію доставки термінових вантажів;
- визначити проблематику та постановку завдання;
- навести цільову та функціональну складову організації доставки термінових вантажів за допомогою безпілотної авіації;
- запропонувати математичні аспекти організації доставки термінових вантажів за допомогою безпілотної авіації;
- розробити алгоритм організації доставки термінових вантажів за допомогою безпілотної авіації;
- навести структуру організації доставки термінових вантажів за допомогою безпілотної авіації;
- відокремити основні модулі системи організації доставки термінових вантажів за допомогою безпілотної авіації;
- провести тестування системи організації доставки термінових вантажів за допомогою безпілотної авіації.

*Об'єкт дослідження є термінові вантажі.*

*Предмет дослідження є процес організації доставки термінових вантажів за допомогою безпілотної авіації.*

*Методи дослідження.* Методологічну і теоретичну основу дослідження склали категорії і принципи науки і техніки, праці вітчизняних і зарубіжних науковців, програмістів. У своєму дослідженні автор застосовує системний, структурно-функціональний методи аналізу, метод математичного моделювання, принципи побудови програмних і апаратних комп'ютерних засобів.

*Практичне значення одержаних результатів.* Використання розробленої моделі дасть можливість підвищити роботу менеджерів з організації доставки термінових вантажів на підприємстві, а тим самим знизить кількість часу витраченого на складання маршруту та розрахунку витрат на перевезення та підвищить якість роботи менеджерів з перевезення на підприємстві за допомогою БПЛА.

*Особистий внесок.* Автором розроблено автоматизовану логістичну систему, щодо завдання у межах дипломної роботи. При розробці даного програмного продукту використовувалися наступні програмні засоби: C++; Qt; Блокнот.

*Результати роботи.* Матеріали дипломного проекту можуть сприяти зменшенню трудомісткості роботи менеджерів з перевезення термінових вантажів на підприємстві, автоматизації їх роботи при контролі та систематизації перевезень на підприємстві за допомогою БПЛА.

Побудова єдиної автоматизованої логістичної системи, що є головним завданням, яке в рамках розвитку процесів інформатизації вирішує кожне сучасне підприємство.

## ЗМІСТ

СПИСОК ВИКОРИСТАНИХ ТЕРМІНІВ, СКОРОЧЕНЬ .....	7
ВСТУП .....	8
Теоретична частина.....	14
1.1    БПЛА: опис, структура .....	15
1.2    Організація доставки термінових вантажів .....	25
1.3    Проблематика та постановка завдання.....	32
Висновки до розділу .....	33
Аналітична частина.....	35
2.1    Цільова та функціональна складова організації доставки термінових вантажів за допомогою безпілотної авіації .....	36
2.2    Математичні аспекти організації доставки термінових вантажів за допомогою безпілотної авіації.....	37
2.3    Алгоритм організації доставки термінових вантажів за допомогою безпілотної авіації .....	67
Висновки до розділу .....	89
Проектна частина .....	90
3.1    Структура організації доставки термінових вантажів за допомогою безпілотної авіації .....	91
3.2    Основні модулі системи організації доставки термінових вантажів за допомогою безпілотної авіації.....	99
3.3    Тестування системи організації доставки термінових вантажів за допомогою безпілотної авіації.....	101
Висновки до розділу .....	109
ВИСНОВКИ.....	110
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	112
ДОДАТКИ.....	117

**СПИСОК ВИКОРИСТАНИХ ТЕРМІНІВ, СКОРОЧЕНЬ**

А	антена
БКУ	бортовий комплекс управління
БПЛА	безпілотні літальні апарати
БСМ	бездротові сенсорні мережі
ДПЛА	безпілотні дистанційно-пілотовані літальні апарати
ЕОМ	електронно-обчислювальна машина
ІЧ	інфрачервоний
К	ключ
КМОП	комплементарна логіка на транзисторах метал-оксид-напівпровідник
ЛП	логарифмічний приймач
НСУ	наземна станція управління
ПЗ	програмне забезпечення
ПК	персональний комп'ютер
ПЛІС	програмована логічна інтегральна схема
ППМ	поворотний пункт маршруту
РТКУ	радіотехнічний комплекс управління
ЦБ	цифровий блок
GPS Global	Positioning System – система глобального позиціонування
PLD	programmable logic device
км	кілометрів
МГц	мегагерц
шт.	штук

# *ВСТУП*

Кафедра ОАП				НАУ 20.02.20. 001 ПЗ			
Виконав	Петрук В.О.			ВСТУП	Літера	Аркуш	Аркушів
Керівник	Юн Г.М.				Д	8	5
Консульт.	Юн Г.М.				ФТМЛ 275 ОП-201М		
Н.Контр.	Жукова С.О.						
Зав. каф.	Юн Г.М.						



**Актуальність дослідження.** Проблеми автоматизованого управління комплексами БПЛА є актуальними в умовах сьогодення, спираючись на ситуацію, що склалася на Сході України. Використання БПЛА в умовах розвідки є пріоритетним напрямком формування стратегічних дій українських військових формувань. Побудова інтелектуальної системи автоматичного управління системою навігації БПЛА здійснюється в якості системи збільшення стійкості для наземного пілота, що передбачає підвищення безпеки польотів та покращення характеристик траєкторії формування шляху БПЛА.

Використання інтелектуальних комплексів в управлінні, реалізує субоптимальний контроль за траєкторією польоту як одного БПЛА так і групи слідкуючих об'єктів.

Методика побудови інтелектуальної системи автоматичного управління системою навігації БПЛА базується на модельно-орієнтованому проектуванні. Для забезпечення польоту БПЛА в умовах перешкод є можливість здійснювати формування характеристик стабільності та керованості на всіх рівнях маневрування.

Головними джерелами вихідної інформації є математична модель динаміки БПЛА, виконуючи пристрої, сенсори, датчики, функції систем орієнтації та кінцеву точку шляху. На основі згаданих параметрів та обмежуючих факторів інтелектуальна система автоматичного управління системою навігації БПЛА повинна сформувати траєкторію шляху.

**Аналіз останніх досліджень і публікацій.** Масштабність сучасних наукових досягнень за темою дослідження є суттєвою. Питання формування інтелектуальних систем управління БПЛА, на сторінках своїх праць, розглядало чимало як зарубіжних так і вітчизняних вчених.

А.Д. Летунов [1], дослідив головні аспекти оптимізації системи управління легкого безпілотного літального апарату за частотним критерієм, математично обґрунтував вибір напрямку руху та обхід перешкод.

Систему інформаційно-вимірювальну та управляючу малорозмірного безпілотного літального апарату підвищеної точності, обґрунтував схемами та розрахунком М. Н. Машнін [2].

Стосовно автоматичного управління системою навігації БПЛА, варто відмітити роботи: М. Н. Красильщикова та Г. Г. Себрякова [3], К. С. Амеліна [4], Ю. Б. Колесова [5], та ін.. Автори детально описують можливість застосування автоматичного управління системою навігації БПЛА в умовах перешкод та механізми формування траєкторії польоту.

Низка авторів сучасників М.Н. Ясечко, А.І. Дохов, М.Г. Іванець, О.В. Тесленко [6] описують, з фізико-математичної точки зору, вплив радіоелектронної апаратури при управлінні БПЛА з відокремленням параметрів сигналів випромінювачів.

Із зарубіжних вчених варто відмітити роботи Williams P., та Crump M [7] та Tang D., Li F., Shen N., і Guo S. [8].

Однак, незважаючи на результати наукових розробок відповідно до теми дослідження, питання формування вдосконаленої методики побудови інтелектуальної системи автоматичного управління системою навігації БПЛА залишається відкритим та потребує детального вивчення.

У масштабі дисертаційних досліджень варто відмітити роботи таких науковців, як: В.А. Полтавський [1], який розкрив методи моделі та системний наліз управління комплексами БПЛА подвійного призначення; А. С. Лучкін [2], який висвітлив основні принципи роботи наземних просторово-розподілених антенних систем радіоліній управління і передачі інформації безпілотних авіаційних комплексів; А.Д. Летунов [3], дослідив головні аспекти оптимізації системи управління легкого безпілотного літального апарату за частотним критерієм, математично обґрунтував вибір напрямку руху та обхід перешкод; М. Н. Машнін [4], навів систему інформаційно-вимірювальну та управляючу малорозмірного безпілотного літального апарату підвищеної точності, обґрунтував дослідження схемами та розрахунком.

Щодо використання БПЛА та управління в умовах перешкод під час проведення воєнних дій, варто відмітити праці таких науковців, як М. Н. Ясечко, А. В. Очкуренко, А. А. Ковальчук, Д. В. Максюта [5], які наводять короткий огляд сучасних засобів придушення радіоелектронної апаратури та перехоплення управління безпілотних літальних апаратів сил антитерористичної операції. Автори детально описують можливість застосування засобу функціонального ураження радіоелектронних систем БПЛА.

І. Іванов та І. Чадов [6] описують питання радіоелектронної боротьби під час воєнних операцій та наводять основні принципи використання механізмів радіоелектронного придушення.

Використання БПЛА в НАТО в умовах локальних конфліктів наводить Ю. Балико [7].

Низка авторів сучасників М.Н. Ясечко, А.І. Дохов, М.Г. Іванець, О.В. Тесленко [8] описують, з фізико-математичної точки зору, вплив радіоелектронної апаратури при управлінні БПЛА з відокремленням параметрів сигналів випромінювачів.

Із зарубіжних вчених варто відмітити роботи David Hastings Dunn [9] та S. Masood [10].

Однак, незважаючи на масштабність наукових розробок відповідно до теми дослідження, питання організації доставки термінових вантажів за допомогою безпілотної авіації залишається відкритим та потребує детального вивчення.

**Мета та завдання дослідження.** Метою роботи є організація доставки термінових вантажів за допомогою безпілотної авіації. Для досягнення поставленої мети у роботі необхідно виконати низку завдань:

- описати сутність та структуру БПЛА;
- дослідити організацію доставки термінових вантажів;
- визначити проблематику та постановку завдання;

- навести цільову та функціональну складову організації доставки термінових вантажів за допомогою безпілотної авіації;
- запропонувати математичні аспекти організації доставки термінових вантажів за допомогою безпілотної авіації;
- розробити алгоритм організації доставки термінових вантажів за допомогою безпілотної авіації;
- навести структуру організації доставки термінових вантажів за допомогою безпілотної авіації;
- відокремити основні модулі системи організації доставки термінових вантажів за допомогою безпілотної авіації;
- провести тестування системи організації доставки термінових вантажів за допомогою безпілотної авіації.

**Об’єкт та предмет дослідження.** Об’єктом роботи є термінові вантажі.

Предметом виступає процес організації доставки термінових вантажів за допомогою безпілотної авіації.

**Методи дослідження.** Методологічну і теоретичну основу дослідження склали категорії і принципи науки і техніки, праці вітчизняних і зарубіжних науковців, програмістів. У своєму дослідженні автор застосовує системний, структурно-функціональний методи аналізу, метод математичного моделювання, принципи побудови програмних і апаратних комп’ютерних засобів.

**Практичне значення одержаних результатів.** Використання розробленої моделі дасть можливість підвищити роботу менеджерів з організації доставки термінових вантажів на підприємстві, а тим самим знизить кількість часу витраченого на складання маршруту та розрахунку витрат на перевезення та підвищить якість роботи менеджерів з перевезення на підприємстві за допомогою БПЛА.

**Особистий внесок.** Автором розроблено автоматизовану логістичну систему, щодо завдання у межах дипломної роботи. При розробці даного

програмного продукту використовувалися наступні програмні засоби: С++; Qt; Блокнот.

**Результати роботи.** Матеріали дипломного проекту можуть сприяти зменшенню трудомісткості роботи менеджерів з перевезення термінових вантажів на підприємстві, автоматизації їх роботи при контролі та систематизації перевезень на підприємстві за допомогою БПЛА.

Побудова єдиної автоматизованої логістичної системи, що є головним завданням, яке в рамках розвитку процесів інформатизації вирішує кожне сучасне підприємство.

**Структура роботи.** Структуру роботи складають: вступ, три розділи, висновки, список використаних джерел, додаток. Загальний обсяг роботи становить 134 сторінки.

## *ТЕОРЕТИЧНА ЧАСТИНА*

Кафедра ОАП				НАУ 20.02.20. 100 ПЗ			
Виконав	Петрук В.О.			Теоретична частина	Літера	Аркуш	Аркушів
Керівник	Юн Г.М.				Д	14	20
Консульт.	Юн Г.М.				ФТМЛ 275 ОП-201М		
Н.Контр.	Жукова С.О.						
Зав. каф.	Юн Г.М.						

## 1.1 БПЛА: опис, структура

Безпілотний літальний апарат (БПЛА, також іноді скорочується як БЛА; в просторіччі іноді використовується назва «безпілотник» (від англ. *Drone* – трутень)) – літальний апарат без екіпажу на борту.

Створений для повітряної зйомки, спостереження за наземними об'єктами і інших завдань в реальному часі.

Розрізняють Безпілотні літальні апарати:

- безпілотні некеровані;
- безпілотні автоматичні;
- безпілотні дистанційно-пілотовані літальні апарати (ДПЛА).

БПЛА прийнято ділити за такими взаємопов'язаними параметрами, як маса, час, дальність і висота польоту: виділяють наступні класи апаратів:

- «Мікро» (умовна назва) – масою до 10 кг, часом польоту близько 1 години і висотою польоту до 1 кілометра;
- «Міні» – масою до 50 кг, часом польоту кілька годин і висотою до 3-5 кілометрів;
- середні ( «міді») – до 1000 кг, часом 10-12 годин і висотою до 9-10 кілометрів;
- важкі – з висотами польоту до 20 кілометрів і часом польоту 24 години і більше.

Ударні БПЛА



Рисунок 1.1 – Пуск AGM-114 Hellfire з ударного БПЛА «Predator»

Безпілотні літальні апарати, здатні нести бойове навантаження і здійснювати знищення наземних, повітряних або над / підводних цілей.

Перші ударні БПЛА з'явилися в 1950-х роках. Це був дистанційно-пілотований протичовновий вертоліт *QH-50 DASH*, зі злітною вагою 1000 кг, оснащений поршнеvim двигуном потужністю 330 л. с. БПЛА міг по командам корабельної системи наведення виходити в район знаходження підводного човна і скидав самонавідну торпеду. Бойовий радіус дії становив 50-70 км. Всього було побудовано більше 750 БЛА, які до 1970 р. стояли на озброєнні ВМС США.

На сьогоднішній день самим поширеними з точки зору застосування є ударні БПЛА, це *MQ-1 Predator* і *MQ-9 Reaper*.

Успішні розробки в цьому напрямі ведуть ізраїльські компанії, російські компанії «Транзас», «ОКБ Сухого».

БПЛА цивільного призначення. Історично склалося так, що початкове застосування БПЛА визначалося як бойовий. Однак з початку 2000-х років колосальне значення стали набувати «мікро-безпілотники», що розробляються не для військових, а суто цивільних цілей [2].



Рисунок 1.2 – Мікро-БПЛА



Громадянська область застосування БПЛА досить велика: від сільського господарства і будівництва до нафтогазового сектора і сектора безпеки [3]. «Дрони» цивільного призначення можуть використовуватися в роботі служб з надзвичайних ситуацій (контроль пожежної безпеки); поліції (патрулювання зон); підприємств сільського господарства (спостереження за посівами), лісництва та рибальства (контроль рибного промислу); компаній, що займаються геодезією (картографування); інститутів географії та геології; компаній нафтогазового сектора (моніторинг нафтогазових об'єктів) [4]; будівельних підприємств (інспектування будівництв); засобів масової інформації (аерофото і відео зйомка) і ін.

Згідно документів, що знаходяться у відкритому доступі документів організацій Європейського Союзу, розподіл споживчого попиту на цивільні БПЛА в період з 2015 по 2020 р виглядає наступним чином: 45% – урядові структури, 25% – пожежні, 13% – сільське господарство і лісництво, 10% – енергетика, 6% – огляд земної поверхні, 1% – зв'язок і мовлення [2].

У світі представлена велика кількість цивільних БПЛА класифікації «мікро» і «міні», що розрізняються за своїми специфікаціям і набору характеристик (призначення, вага, розмір, тривалість і висота польоту, система запуску і приземлення, наявність систем автопілотування і навігації, формат фото- і відеозйомки і ін.).

Найбільшим у світі БПЛА є багаторазовий космічний корабель Буран, з поправкою, що зліт здійснюється за допомогою зовнішньої енергії (ракети), однак і свехмалі БПЛА запускаються точно так же, носієм і джерелом стартової енергії виступає людина.

Для визначення координат і земної швидкості сучасні БПЛА, як правило, використовують супутникові навігаційні приймачі (*GPS* або *ГЛОНАСС*). Кути орієнтації і перевантаження визначаються з використанням гіроскопів і акселерометрів.

У якості керуючої апаратури, як правило, використовуються спеціалізовані обчислювачі на базі цифрових сигнальних процесорів або комп'ютери формату PC / 104, *MicroPC* під управлінням операційних систем реального часу (*QNX, VME, VxWorks, XOberon*). Програмне забезпечення пишеться зазвичай на мовах високого рівня, таких як Сі, Сі ++, Модула-2, Оберон SA або Ада95.

Для передачі на пункт управління відеоданих, отриманих з бортових сенсорів, в складі БПЛА є радіопередавач, що забезпечує радіозв'язок з прийомним устаткуванням. Залежно від формату зображень і ступеня їх стиснення пропускна здатність цифрових радіоліній передачі даних може становити одиниці-сотні Мбіт/с [5-6]. Крім того, до складу бортових радіозасобів БПЛА повинні входити приймач команд управління, а також передавач службової (телеметричної) інформації.

Початок проектування засобів, подібних до сучасних БПЛА припадає на 1899 рік. У 1899 році Нікола Тесла розробив і продемонстрував мініатюрне радіокерований судно.



Рисунок 1.3 – БПЛА Ту-141 «Стриж» розробки ОКБ Туполєва, 1974 рік

У 1910 році, натхненний успіхами братів Райт, молодий американський військовий інженер з Огайо Чарльз Кеттерінг запропонував використовувати літальні апарати без людини. За його задумом керований годинниковим механізмом пристрій в заданому місці повинен був скидати крила і падати, як

бомба, на ворога. Отримавши фінансування армії США, він побудував і з перемінним успіхом випробував кілька пристроїв, які отримали назви *The Kattering Aerial Torpedo*, *Kettering Bug*, але в бойових діях вони так і не застосовувалися. У Німеччині розробляється проект радіокерованого безпілотного бомбардувальника *Feldermaus*.

У 1933 році у Великобританії розроблений перший БПЛА багаторазового використання «*Queen Bee*». Були використані три відреставрованих біплана *Fairy Queen*, дистанційно керовані з судна по радіо. Два з них зазнали аварії, а третій зробив успішний політ, зробивши Великобританію першою країною, що отримала користь з БПЛА. Ця радіокерована безпілотна мішень під назвою *DH82A Tiger Moth* використовувалася на королівському Військово-морському флоті з 1934 по 1943 р Армія і ВМФ США з 1940 року використовували БПЛА *Radioplane OQ-2* в якості літака-мішені.

Протягом Другої світової війни німецькі вчені вели розробки декількох радіокерованих типів зброї, включаючи керовані бомби *Henschel Hs 293* і *Fritz X* (англ. *Fritz X*), ракету *Enzian* (англ. *Enzian*) і радіокерований літак, наповнений вибуховою речовиною. Незважаючи на незавершеність проектів, *Fritz X* і *Hs 293* з успіхом використовувалися на Середземному морі проти броньованих військових кораблів. Масовою зброєю була перша «крилата ракета» Фау-1 з реактивним пульсуючим двигуном, яка могла запускатися як з землі, так і з повітря. У нацистській Німеччині в 1942 році було запущено виробництво ракет Фау-2, що мають систему управління, яка утримує ракету на заданій при старті траєкторії протягом усього польоту. Були розроблені і застосовувалися керовані авіабомби.

В СРСР в 1930-1940 рр. авіаконструктором Нікітіним розроблявся торпедоносець – планёр спеціального призначення (ПСН-1 і ПСН-2) типу «літаюче крило» в двох варіантах: пілотований тренувально-пристрілювальний і безпілотний з повною автоматикою. До початку 1940 року був представлений проект безпілотної літаючої торпеди з дальністю

польоту від 100 км і вище (при швидкості польоту 700 км / год). Однак цих розробкам не було судилося втілитися в реальні конструкції. У 1941 році були вдалі застосування важких бомбардувальників ТБ-3 в якості БПЛА для знищення мостів.



Рисунок 1.4 – *Luna X-2000*

У США запустили в масове виробництво БПЛА-мішень *Radioplane OQ-2* для тренування льотчиків і зенітників. Також, в 1944 році був застосований вперше в світі класичний ударний БПЛА – *Interstate TDR*. Крім цього, військовими США був створений цілий ряд керованих авіабомб, включаючи найбільш досконалу технічну зброю, застосовану в роки війни – самонавідну планувальну бомбу *ASM-N-2 Bat*, перше в світі зброю схеми «вистрілив-і-забув», що не вимагає втручання оператора. Після війни розробки безпілотних літальних апаратів в США тимчасово змістилися в бік створення керованих ракет і авіабомб, лише в 1960-х повернувшись до ідеї не ударних БПЛА.

У СРСР 23 вересня 1957 року КБ Туполева отримало держзамовлення на розробку мобільної ядерної надзвукової крилатої ракети середнього

радіусу дії. Перший зліт моделі Ту-121 був здійснений 25 серпня 1960 року, але програма була закрита на користь балістичних ракет КБ Корольова. Створена ж конструкція знайшла застосування в якості мішені, а також при створенні безпілотних літаків розвідників Ту-123 «Яструб», Ту-143 «Рейс» і Ту-141 «Стриж», що стояли на озброєнні ВПС СРСР з 1964 по 1979 рік. Ту-143 «Рейс» протягом 1970-х років поставлявся в африканські і близькосхідні країни, в тому числі і в Ірак. Ту-141 «Стриж» складається на озброєнні ВПС України і понині. Комплекси «Рейс» з БПЛА Ту-143 експлуатуються до теперішнього часу, поставлялися в Чехословаччину (1984), Румунію, Ірак і Сирію (1982), використовувалися в бойових діях під час Ліванської війни. У Чехословаччині в 1984 році були сформовані дві ескадрильї, одна з яких в даний час знаходиться в Чехії, інша – в Словаччині[7].

СРСР ще в 1970-ті – 1980-ті роки був лідером по виробництву БПЛА – тільки Ту-143 було випущено близько 950 штук.

На початку 1960-х років дистанційно-пілотовані літальні апарати використовувалися США для стеження за розміщеннями ракет на Кубі і в Радянському Союзі – після того, як були збиті RB-47 і два U-2, для виконання розвідувальних робіт було розпочато розробку висотного безпілотного розвідника *Red Wadon* (модель 136). БПЛА мав високо розташовані крила і малу радіолокаційну і інфрачервону помітність.

Під час війни у В'єтнамі, з ростом втрат американської авіації від ракет в'єтнамських ЗРК, зросло використання БПЛА. В основному вони використовувалися для ведення фоторозвідки, іноді для цілей РЕБ. Зокрема, для ведення радіотехнічної розвідки застосовувалися БПЛА 147Е. Незважаючи на те, що, в кінцевому рахунку, безпілотник був збитий, він передавав на наземний пункт характеристики радянського ЗРК С-75 протягом всього свого польоту, і цінність цієї інформації була порівнянна з повною вартістю програми розробки безпілотного літального апарату. Вона також дозволила зберегти життя багатьом американським льотчикам, а також літаки протягом наступних 15 років, аж до 1973 року.

В ході війни американські БПЛА зробили майже 3500 польотів, причому втрати склали близько чотирьох відсотків. Апарати застосовувалися для ведення фоторозвідки, ретрансляції сигналу, розвідки радіоелектронних засобів, РЕБ і в якості помилкових цілей для ускладнення повітряної обстановки. Але повна програма БПЛА була оповита таємницею настільки, що її успіх, який повинен був стимулювати розвиток БПЛА після кінця військових дій, в значній мірі залишився непоміченим.



Рисунок 1.5 – Розвідувальний БПЛА «виріб 61» «Бджола» ОКБ Яковлєва, 1990 рік

Безпілотні літальні апарати застосовувалися Ізраїлем під час арабо-ізраїльського конфлікту в 1973 р. Вони використовувалися для спостережень і розвідки, а також в якості помилкових цілей. У 1982 році БПЛА використовувалися під час бойових дій в долині Бекаа в Лівані. Ізраїльський БПЛА *IAI Scout* і малорозмірні дистанційно-пілотовані літальні апарати *Mastiff* провели розвідку і спостереження сирійських аеродромів, позицій ЗРК і пересувань військ. За інформацією, отриманою за допомогою БПЛА, відволікаюча група ізраїльської авіації перед ударом головних сил викликала включення радіолокаційних станцій сирійських ЗРК, за якими було завдано удару з допомогою самонавідних протирадіолокаційних ракет, а ті кошти, які

не були знищені, були придушені перешкодами. Успіх ізраїльської авіації був вражаючим – Сирія втратила 18 батарей ЗРК.

Дистанційно-пілотовані літальні апарати і автономні БПЛА використовувалися обома сторонами протягом війни в Перській затоці 1991 роки (операція « Буря в пустелі»), перш за все як платформи спостереження і розвідки. США, Великобританія, і Франція розгорнули і ефективно використовували системи типу *Pioneer, Pointer, Exdrone, Midge, Alpilles Mart, CL-89*. Ірак використовував *Al Yamamah, Makareb -1000, Sahreb-1 i Sahreb-2*.

Під час цієї операції БПЛА тактичної розвідки коаліції зробили більше 530 вильотів, наліт склав близько 1700 годин. При цьому 28 апаратів були пошкоджені, включаючи 12, які були збиті. З 40 БПЛА *Pioneer*, використовуваних США, 60% були пошкоджені, але 75% виявилися ремонтпридатними. З усіх втрачених БПЛА тільки 2 відносилися до бойових втрат. Низький коефіцієнт втрат обумовлений, найімовірніше, невеликими розмірами БПЛА, в силу чого іракська армія визнала, що вони не становлять великої загрози.

БПЛА також використовувалися і в операціях з підтримання миру силами ООН в колишній Югославії. У 1992 році Організація Об'єднаних Націй санкціонувала використання військово-повітряних сил НАТО, щоб забезпечити прикриття Боснії з повітря, підтримувати наземні війська, розміщені по всій країні. Для виконання цього завдання потрібно ведення цілодобової розвідки.

У 1992 році ізраїльський БПЛА був вперше використаний як бойовий засіб для цілевказівки при операції по ліквідації в південному Лівані лідера терористичної організації Хезболла Аббаса аль-Мусаві. БПЛА вистежив колону, в якій їхав Мусаві і позначив його автомобіль лазерним маркером, за яким була випущена ракета зі штурмового вертольота.

Під час Інтифади аль-Акса в 2000-2005 році ВПС Ізраїлю широко використовували БПЛА в комплексі: для спостереження, розвідки, цілевказівки. Широке застосування отримали ударні БПЛА для точкових

ліквідацій терористів, особливо в секторі Газа, захопленому терористичною організацією ХАМАС.

БПЛА дальнього радіусу дії використовувалися Ізраїлем в ході ряду операцій з перехоплення і знищення контрабандних поставок Іраном озброєнь палестинським терористам, в числі яких перехоплення судна Карін А з 50 тоннами зброї в Червоному морі в 2002 році, Франкоп з 500 тоннами зброї в Середземному морі в 2009 році, і багато подібних.

Також у 2009 році в Судані сталася серія повітряних атак не ідентифікованими літальними апаратами на транспорти і склади іранської зброї, адресованої на Синай і в сектор Газа. Припущення, що атаки були здійснені ізраїльськими ударними БПЛА дальнього радіусу дії не були підтверджені ізраїльськими офіційними джерелами – традиційно для таких подій.

У квітні 2015 року БПЛА Х-47В здійснив першу в історії процедуру дозаправки в повітрі повністю в автоматичному режимі [8].



## 1.2 Організація доставки термінових вантажів

До термінових вантажів варто віднести кров, біологічний матеріал, вакцини, медичні препарати. Особливості транспортування та зберігання яких є особливими та потребують спеціальних засобів.

При транспортуванні і зберіганні медико-біологічної продукції повинні бути дотримані умови, що забезпечують збереження вихідної якості препаратів; препарати повинні бути забезпечені від впливу змінюваної температури навколишнього середовища, від руйнування та пошкодження упаковок і т.д.

Транспортування і зберігання повинно проводитися при дотриманні спеціальної системи "холодового ланцюга".

Холодовий ланцюг – це безперебійно функціонуюча система, яка забезпечує оптимальний температурний режим зберігання і транспортування вакцин та інших імунобіологічних препаратів на всіх етапах шляху їх слідування від підприємства-виготовлювача до людини, якій здійснюють вакцинацію.

Оптимальною для зберігання і транспортування вакцин та інших імунобіологічних препаратів є температура у межах 0- + 8 ° С. У разі необхідності тривалого зберігання живих вірусних вакцин (проти кору, паразита і поліомієліту) рекомендується їх зміст в замороженому вигляді, тобто при температурі -20 ° С. Категорично не допускається заморожування адсорбованих препаратів (вакцин АДС, АКДС і ін.). При визначенні режиму зберігання і транспортування інших вакцин слід керуватися інструкціями, що додаються до препарату.

Основними компонентами холодового ланцюга є:

- спеціально навчений персонал, що забезпечує обслуговування холодильного обладнання, правильне зберігання вакцин і постачання їх до структурних підрозділів;

- холодильне обладнання, призначене для зберігання і транспортування вакцин в оптимальних температурних умовах;
- механізм контролю за дотриманням необхідного температурного режиму на всіх етапах холодового ланцюга.

В системі холодового ланцюга є такі рівні

1-й рівень - підприємство-виробник вакцин та інших імунобіологічних препаратів;

2-й рівень - крайові, обласні аптечні склади і склади;

3-й рівень - міські та районні (міські та сільські) аптечні склади;

4-й рівень - лікувально-профілактичні заклади (дільничні лікарні, амбулаторії, дитячі поліклініки, пологові будинки, фельдшерсько-акушерські пункти та ін.).

На всіх рівнях холодового ланцюга повинна строго проводитися реєстрація надходжень і подальшого відправлення вакцин та інших імунобіологічних препаратів в нижчестоящі установи з фіксацією найменування препарату, його кількості та серії, терміну придатності, дати надходження (відправлення), умов зберігання та транспортування, показників термоіндикаторів, прізвище та ініціали відповідального працівника. Крім того, на всіх рівнях холодового ланцюга необхідно регулярно (не рідше 2 разів на добу) в спеціальному журналі відзначати температуру, при якій зберігається той чи інший препарат із зазначенням відповідальної особи за цю роботу.

На всіх етапах холодового ланцюга повинні працювати спеціально призначені в наказовому порядку фахівці з чітким визначенням їх службових обов'язків відповідно до займаної посади і об'ємом роботи.

У кожній установі, де зберігаються імунобіологічні препарати, незалежно від їх кількості, повинен бути затверджений його керівником план екстрених заходів на випадок виникнення непередбачених проблем у холодового ланцюга в цій установі.

При зберіганні вакцин слід дотримуватися ряду загальних правил:

- вакцина (вакцини) повинна розташовуватися таким чином, щоб до кожної її упаковці був доступ повітря;
- вакцина (вакцини) повинна розташовуватися так, щоб препарат, який має менший термін придатності, використовувався в першу чергу;
- в холодильниках (при температурі  $0 + 8^{\circ}\text{C}$  слід зберігати БЦЖ і інші неадсорбовані і адсорбовані вакцини. Живі вакцини, які відповідно до інструкції по застосуванню вимагають зберігання в замороженому стані, повинні зберігатися в морозильній камері при температурі  $-20^{\circ}\text{C}$ . Для таких препаратів допускається тимчасове, не більше 48 год, підвищення температури до  $0- + 8^{\circ}\text{C}$  при транспортуванні.

Перший рівень холодового ланцюга

Першим рівнем холодового ланцюга є підприємство-виробник, незалежно від свого територіального розташування.

На тому рівні вакцина до відправки повинна зберігатися:

- живі вірусні (корова, паротина, поліомієлітна і ін.) при температурі не вище  $-20^{\circ}\text{C}$ , тобто в замороженому вигляді;
- інші вакцини при температурі  $0 + 8^{\circ}\text{C}$  (відповідно до затвердженої інструкції щодо їх застосування).

Для зберігання вакцин на цьому рівні, в залежності від виду та кількості препарату, можуть використовуватися морозильні і холодильні камери, морозильники та холодильники, холодильні шафи.

Відправка вакцин та інших імунобіологічних препаратів повинна здійснюватися в термоконтейнерах в супроводі відповідного термоіндикатора <\*> (10).

Термоконтейнер є ящик, виготовлений з полістиролу або поліуретану з кришкою з того ж матеріалу, що герметично закривається. Зазвичай цей ящик поміщається в картонну коробку, що оберігає його від швидкого псування. На коробку наноситься кольорова етикетка: "Вакцина! Терміновий

вантаж!", "Боїться заморожування!". Температура в термоконтейнерах, яка досягається завдяки приміщенню в його порожнину холодоелементів, повинна зберігатися протягом мінімум 48 год в межах від 0 до + 8 ° С при температурі зовнішнього середовища + 43 ° С Розміри термоконтейнера можуть варіювати.

Холодоелементами є пластмасова ємність прямокутної форми, що герметично закривається пробкою. Холодоелементи бувають об'ємом від 0,3 до 0,6 л.

Підприємство-виробник повинно завжди мати в своєму розпорядженні достатній запас термоконтейнерів і холодоелементів до них, а також термоіндикаторів.

Підприємство-виробник повинно відправляти вакцини таким чином, щоб термін їх придатності дозволяв використання препаратів протягом, принаймні, 3-х місяців з моменту їх надходження в аптечні склади і склади крайового і обласного рівнів.

Відправка вакцин повинна здійснюватися через найближчі аеропорт або залізничну станцію, куди препарати, поміщені в термоконтейнер, повинні доставлятися у критому автотранспорті в умовах, що виключають механічне пошкодження вакцин і вплив на них несприятливих температурних факторів.

Тривалість транспортування вакцини від підприємства-виготовлювача до 2-го рівня холодового ланцюга не повинна перевищувати 48 год. Виходячи з цього повинен проводитися вибір відповідного транспортного засобу. Забороняється відправлення вакцини транспортом, прибуття якого в пункт призначення припадає на неробочі дні, за винятком випадків, коли замовник письмово гарантує отримання вакцини в такі дні.

Відповідне транспортне агентство повинно надавати транспортуванні вакцин та інших імунобіологічних препаратів найбільш сприятливі умови, включаючи скорочення термінів відправки і доставки, виняток впливу несприятливих фізичних і термічних факторів як в період перебування

препаратів в аеропорту (залізничної станції), в тому числі і транзитом, так і в період їх транспортування.

У разі затримки відправлення вакцини, останню слід помістити в холодильні камери або шафи, якщо такі є в аеропорту (залізничної станції). У разі відсутності таких і при значній затримці відправлення (до 12 год і більше) вакцини необхідно повернути на підприємство-виробник за домовленістю з останнім.

Одержувач вакцин повинен завчасно узгодити з відповідним транспортним агентством на місці процедуру отримання вантажу безпосередньо з борту літака або вагона поїзда.

Доставка отриманої вакцини до аптечного складу 2-го рівня холодового ланцюга здійснюється в максимально стислі терміни.

Транспортування здійснюється в суворій відповідності з супровідними документами.

Другий рівень холодового ланцюга: крайові та обласні аптечні склади.

При отриманні вакцини з аеропорту (залізничної станції) або доставці її авторефрежираторним транспортом необхідно негайно помістити вантаж в холодильну камеру, де провести розвантаження термokonтейнерів. При цьому необхідно перевірити показники відповідних термоіндикаторів і внести в них необхідні записи, а також занести всі регламентовані дані в журнал отримання і відправки вакцин.

Вивільнені термokonтейнери назад на підприємства не направляються. Вони або знищуються, або використовуються на місці по розсуд особи, відповідальної за холодовий ланцюг на цьому рівні.

На цьому рівні холодового ланцюга всі вакцини зберігаються в холодильній камері, холодильних шафах або холодильниках при температурі  $0 + 8^{\circ} \text{C}$ . У випадках, коли виникає необхідність зберігання препаратів більш 3-х місяців, живі вірусні вакцини рекомендується зберігати в межах терміну придатності в замороженому стані, тобто в морозильних камерах і морозильниках.

Особа, відповідальна за холодний ланцюг на цьому рівні, має мати заздалегідь узгоджений з нижчими структурами чіткий план постачання вакцинами відповідних установ районного і міського рівня (3-й рівень холодного ланцюга). На 2-му рівні холодного ланцюга, тобто на рівні аптечних складів крайового і обласного рівнів, транспортування вакцин та інших імунобіологічних препаратів також має здійснюватися в термоконтейнерах.

Аптечний склад або склад ЦГСЕН цього рівня повинен завжди мати достатній запас заморожених холодоелементами і резервний запас термоконтейнерів для транспортування вакцин та інших імунобіологічних препаратів в установи третього рівня холодного ланцюга.

Завантаження вакцини (вакцин) в термоконтейнер (и) повинна здійснюватися в холодильній камері. У виняткових випадках завантаження може проводитися при кімнатній температурі, але в гранично стислі терміни (не більше 5 - 10 хв).

Особа, відповідальна за холодний ланцюг на цьому рівні, здійснює контроль функціонування холодного ланцюга на третьому рівні.

Третій рівень холодного ланцюга: міські та районні (Сільські та міські) аптечні склади.

Всі види вакцин на цьому рівні повинні зберігатися при температурі в межах 0- + 8 ° С.

Для зберігання вакцин на цьому рівні холодного ланцюга аптечні склади повинні мати у своєму розпорядженні холодильники або холодильні шафи з достатнім корисним об'ємом, що дозволяє зберігати необхідне для території, що обслуговується кількість вакцин. Крім того для заморожування необхідної кількості холодоелементами необхідний морозильник (морозильники).

Необхідно мати достатню кількість термоконтейнерів для транспортування необхідної кількості вакцин з аптечного складу краю або області, а в разі потреби і в деякі установи, що здійснюють

вакцинопрофілактику. В останньому випадку використаний термоконтейнер повинен бути повернутий разом з холодоелементами власнику (міський або районний аптечний склад).

Транспортування вакцин з вищого рівня холодового ланцюга здійснюється авто- або іншим доступним видом транспорту (БПЛА) в термоконтейнерах.

Особа, відповідальна за холододовий ланцюг на цьому рівні, має вести строгий облік надходжень і відправлень вакцин, регулярно фіксувати показники температурного режиму на складі, показники термоіндикаторів.

Особа, відповідальна за холододовий ланцюг на цьому рівні контролює функціонування холодового ланцюга в установах, що здійснюють вакцинопрофілактику.

Четвертий рівень холодового ланцюга: лікувально-профілактичні установи, які здійснюють вакцинопрофілактику

Всі види вакцин на цьому рівні повинні зберігатися в холодильній камері при температурі  $0 + 8^{\circ} \text{C}$  в суворій відповідності до інструкцій із застосування препаратів.

Розчинник для вакцин повинен зберігатися також в холодильнику, щоб в разі підготовки вакцини для застосування не викликати підвищення температури останньої. У морозильній камері необхідно мати запас заморожених холодоелементів. тривалість зберігання вакцини на цьому рівні холодового ланцюга не повинна перевищувати 1 місяця.

На цьому рівні холодового ланцюга необхідно мати холодильник (холодильники), призначений тільки для зберігання вакцин і інших імунобіологічних препаратів. Зберігання в них інших препаратів або предметів не допускається. Тип холодильника визначається виходячи з корисного обсягу його холодильної камери і кількості необхідних вакцин для вакцинації населення території, що обслуговується.

Установа повинна мати достатню кількість термоконтейнерів для транспортування вакцин з міського або районного аптечного складу, а такі

для проведення вакцинації на виїзді. Обсяг термоконтейнерів залежить від обсягу проведеної цієї установою по вакцинації населення відповідної території. Необхідно також мати достатню кількість холодоелементів. Транспортування вакцин здійснюється авто- або іншим видом транспорту. Розвантаження чи завантаження термоконтейнерів повинно здійснюватися максимально швидко, не більше 5 - 10 хв.

Особа, відповідальна за холододовий ланцюг на цьому рівні, має вести строгий облік надходження та витрати вакцин та інших імунобіологічних препаратів, фіксувати показники температурного режиму зберігання препаратів і показники термоіндикаторів.

### **1.3 Проблематика та постановка завдання**

У багатьох країнах, що розвиваються доступ до життєво важливих медичних препаратів, від яких часом залежить життя, утруднений у зв'язку з відомою проблемою доставки «на останній милі». Йдеться про неможливість доставити необхідний препарат з міста в сільську місцевість або віддалені райони через відсутність належної транспортної інфраструктури, а також інфраструктури зв'язку та ланцюгів поставок.

У багатьох віддалених місцевостях головною причиною смертності жінок після пологів є післяродова геморрагія. Кров необхідно зберігати і перевозити при певній температурі, інакше вона швидко стає непридатною для застосування. Оскільки номенклатура препаратів крові досить широка і точно розрахувати майбутні потреби в них неможливо, багато клінік переливання крові не зберігають достатніх запасів крові на всі випадки життя.

Протягом тривалого сезону дощів багато доріг розмиваються і стають непрохідними або зовсім зникають. В результаті часто буває так, що люди, які потребують рятівного переливання крові, не можуть її отримати і гинуть.



На сьогодні, актуальним питанням є впровадження механізмів доставки вантажів за допомогою БПЛА дозволяє клінікам з переливання крові в західній частині країни розміщувати термінові замовлення на доставку крові шляхом відправки SMS-повідомлень зі стільникових телефонів.

Дальність польоту БПЛА в обидва кінці навіть у вітряну і дощову погоду сягає 150 км. Апарат може переносити 1,5 кг крові – цієї кількості достатньо для порятунку людського життя. Безпілотник знижується до невеликої висоти і скидає вантаж на позначене місце поруч з відповідним медичним центром, яке називається «поштова скринька», а після виконання доставки повертається в «Гніздо». БПЛА можуть здійснювати від 50 до 150 екстрених вильотів на добу і доставляти вантажі в 21 клініку переливання крові. Вони дозволяють виконувати замовлення приблизно за 30 хвилин. Також по мимо крові є можливість перевозити вакцини та медичні препарати, які є життєво важливими.

Метою даної роботи є організація доставки термінових вантажів за допомогою безпілотної авіації. У роботі пропонується розробити програмний додаток для контролю завантаження та прорахунку відстані польоту для БПЛА в умовах доставки термінових вантажів.

### **Висновки до розділу**

У межах першого розділу здійснено дослідження теоретико-методологічних аспектів за темою дослідження, розкрито поняття БПЛА та основні різновиди. Здійснено опис поняття терміновий вантаж, описано актуальність перевезення термінових вантажів та запропоновано завдання для подальшого дослідження.

БПЛА можуть докорінно змінити способи доставки медикаментів для надання невідкладної допомоги у віддалені райони. Час, яке ця технологія дозволяє економити на доставці препаратів крові або вакцини людині, яка заразилася сказом, може виявитися вирішальним для порятунку його життя.

Впровадження цієї технології може стати значущим кроком до того, щоб забезпечити такими препаратами всіх дітей та людей, які страждають на хвороби.

## *АНАЛІТИЧНА ЧАСТИНА*

Кафедра ОАП				НАУ 20.02.20. 200 ПЗ			
Виконав	Петрук В.О.			Аналітична частина	Літера	Аркуш	Аркушів
Керівник	Юн Г.М.				Д	35	52
Консульт.	Юн Г.М.				ФТМЛ 275 ОП-201М		
Н.Контр.	Жукова С.О.						
Зав. каф.	Юн Г.М.						

## **2.1 Цільова та функціональна складова організації доставки термінових вантажів за допомогою безпілотної авіації**

Для розробленої системи організації доставки термінових вантажів за допомогою безпілотної авіації необхідні відповідні апаратні засоби: фірмове програмне забезпечення, спеціальний персонал, який підтримував би роботу інформаційної системи.

Впровадження автоматизованої системи організації доставки термінових вантажів за допомогою безпілотної авіації дозволяє оперативно враховувати важливі параметри роботи підприємства з перевезень термінових вантажів, звести до мінімуму втрати часу і транспортні витрати, а значить, знизити ціни на перевезення. Автоматизована система організації доставки термінових вантажів за допомогою безпілотної авіації повинна бути недорогою, простою в обслуговуванні, розрахованою на роботу одного або двох чоловік. Інформаційна система, яка передбачає врахування таких процедур, є документально-фактографічною. Так як основним документом, що відображає факт діяльності, є подорожній лист, він повинен відображати факти господарської діяльності у вигляді наступних відомостей.

- Складання маршрутів по замовленнях;
- Контроль схеми замовлення (перевезення);
- Обрання технічних засобів для здійснення замовлення;
- Обирання критерію оптимізації;
- Складання маршруту за перевезення;
- Формування маршруту.

## 2.2 Математичні аспекти організації доставки термінових вантажів за допомогою безпілотної авіації

Для того, щоб спланувати шляхи групи БПЛА, які з'єднують вибрані точки, обмеження повинні бути прийняті до уваги. Основним чинником, обмеження є кривизна або радіус розвороту. Це, разом з плануванням найкоротшого шляху, вимагає розробки підходів, які прийматимуть це обмеження в розрахунок.

Досить часто, вигідно мати короткі шляхи, для того, щоб звести до мінімуму час прольоту, а також зниження витрата палива, і пов'язаний з перевагою мінімум енергії і потужності.

Методи можуть бути розширені для отримання короткого шляху з більш довгих маршрутів з метою обходу перешкод або для синхронізації часу прибуття автономних транспортних засобів. Алгоритми для проектування найкоротших маршрутів проводяться в різних полях, таких як обчислювальна геометрія, дослідження операцій та логістика. Однією з добре досліджуваних проблем в обчислювальній геометрії є те, що про знаходження найкоротшого маршруту було відомо, ще у 1987.

Завдання комівояжера (TSP) і проблема китайського листоноші (CPP), широко вивчаються в галузі дослідження операцій, але вони будуються у вигляді графа і рішень не мають.

На сьогодні, актуальними є три види шляхів: (I) Dubins – шлях з дугами постійної кривизни; (II) клотоїда – шляхи, які схожі з Dubins шляхами, але зі змінною кривизни дуг; і (III) Піфагора – годограф (PH) шляху.

Dubins шлях – це найкоротший шлях, але він має переривчасту кривизну. Решта два типи шляхів є безперервної кривизни, але їх складніше обчислити. У першому випадку, дуги постійної кривизни замінюються клотоїдами дуг, які забезпечують плавний перехід кривизни між лінійними сегментами шляху. Крива PH, в даній кривій параметризовані поліноми в

залежності від довжини шляху і виконана вона у вигляді однієї кривої, а не як набір з'єднаних сегментів.

<b>Характеристики</b>	<b>Величина</b>	<b>Примечание</b>
<b>Взлетный вес макс. кг</b>	<b>300</b>	
<b>Размах крыла, ММ</b>	<b>5800</b>	
<b>Вес пустого , кг</b>	<b>150</b>	
<b>Длина , мм</b>	<b>2400</b>	
<b>Высота на земле, мм</b>	<b>1700</b>	
<b>Ёмкость кессонных баков, л</b>	<b>90</b>	
<b>Крейсерская скорость, км/ч</b>	<b>280 -300</b>	
<b>Длина разбега-пробега, м</b>	<b>350-400</b>	
<b>Время в полете, час</b>	<b>6-12</b>	
<b>Часовой расход топлива, л/ч</b>	<b>14-15</b>	
<b>Комерческая нагрузка, кг</b>	<b>150</b>	<b>В контейнерах</b>
<b>Стоимость серийного БПЛА</b>	<b>50-60 тис</b>	
<b>Двигатель бензиновый, л/с</b>	<b>64</b>	

Рисунок 2.1 – Характеристика БПЛА який використовується від Києва до районного центру

Dubins Шляхи. У своїй найпростішій формі, шлях складається з комбінації лінійного сегмента і сегментів постійної кривизни. Поєднання обох цих прямолінійних і кругових дуг буде виробляти короткий маневр для одного БПЛА між двома іншими у просторі. На практиці, така поведінка може бути визначена між двома або більше точками у просторі, щоб

визначити шляхи прольоту для БПЛА. У разі автономного транспортного засобу, на додаток до пунктів, орієнтація транспортного засобу також має важливе значення. Наприклад, це може бути, якщо необхідно задати кут підходу до об'єкта для кращого огляду.

Таким чином, важливо, щоб найкоротший шлях знаходився між стартовою позицією з орієнтацією і положенням кінцевої позиції з орієнтацією.

Найкоротший шлях, що з'єднує дві позиції є шлях Dubins. Математичний доказ надається Dubins (1957). Найпростіше визначення стосовно шляху Dubins виглядає наступним чином: найкоротший шлях, який відповідає максимальній кривизні пов'язаний між двома точками з специфічною орієнтацією у площині або CLC або шлях CCC, або підмножина з них, де C являє собою дугу кола і L являє собою пряму лінію дотичну до C.

Shkel і Lumelsky (2001), позначають наступне визначення: довжини CLC шляхів розраховуються для заданого набору поз. CLC шлях формується шляхом з'єднання двох дуг кіл лінією, дотичною до них, і шлях CCC складається з трьох послідовних дуг які дотичні один до одного, у підмножини цих шляхів CL, LC і CC.

Рисунок 6.1 показує шляхи CLC і CCC. Додаток, який включає в себе відстеження цілі тісно вивчається з використанням шляху CCC в Вонг та ін. (2004).

Таким чином, дослідивши основні положення, зосередимося на CLC і CC шляхах, і, хоча є вибір методів, щоб довести, що шлях Dubins є коротким, література не визначає прості способи отримання шляху Dubins. Враховуючи це, орієнтуємося на виробництві шляху Dubins, використовуючи принципи аналітичної та диференціальної геометрії.

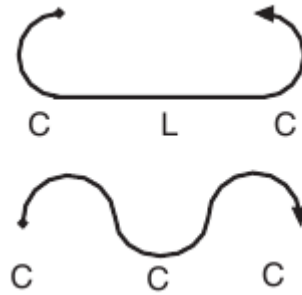


Рисунок 2.2 – CLC і CCC типи Dubins шляху

Найкоротший шлях групи БПЛА формується шляхом вибору двох дотичних кругових дуг. Початкові і кінцеві позиції повинні лежати на дузі, радіус дуги визначено радіусами кривизни, отриманої від радіусу повороту транспортних засобів, і центрами дуг, що визначається з центрами кривизни. Таким чином, завдання зводиться до визначення положення загальних дотичних дуг між двома дугами. Як зазначалося раніше, існує два загальних дотичних кола:

- (I) внутрішнє / внутрішнє дотичні,
- (II) зовнішнє / зовнішнє дотичні.

Перше паралельно лінії, що з'єднує центри дуг для того ж радіуса, а друге перетинає центральне. Якщо початкового та кінцевого кола дуги перетинаються, це можливо допустити тільки в тому випадку, якщо зовнішнє дотичне рішення, як внутрішнє дотичне не існує.

Для поставленого завдання є чотири можливі рішення. Рисунок 2.2 показує, що для даної пози, БПЛА може або повернутий праворуч або повернутий ліворуч. Таким чином, шлях може початися і закінчуватися або за годинниковою стрілкою (вправо) або проти годинникової стрілки (ліворуч).

Фігура показує, що, так як є два оберти для даної пози, то є дві окружності, одна, у свою чергу, позначена R на дузі C1 та інші з лівого боку



позначені L на дузі C2. Таким чином, пара поз виробляє чотири Dubins шляхи.

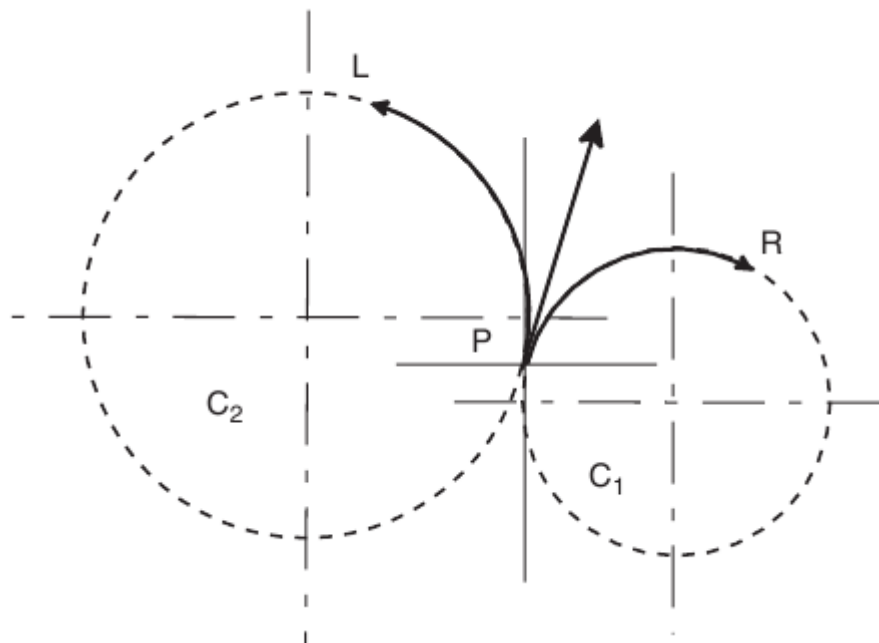


Рисунок 2.3 – Окружності

Рисунок 2.3 та 2.4 відповідно показані чотири Dubins шляхи як із зовнішніми так із внутрішніми дотичними дугами. За для проектування шляху Dubins потрібні наступні вхідні параметри.

- (I) Початок представляють: PS (XS, YS,  $\phi_s$ ).
- (II) Оздоблення представляють: P (XF, YF,  $\phi_f$ ).
- (III) Запуск кривизни:  $k_s \leq k_{\max}$ , де  $k_s = 1 / \rho_s$  і  $\rho_s$  є радіус кривизни.
- (IV) Оздоблення кривизни:  $k_f \leq k_{\max}$ , де  $k_f = 1 / \rho_f$  і  $\rho_f$  є радіус кривизни,

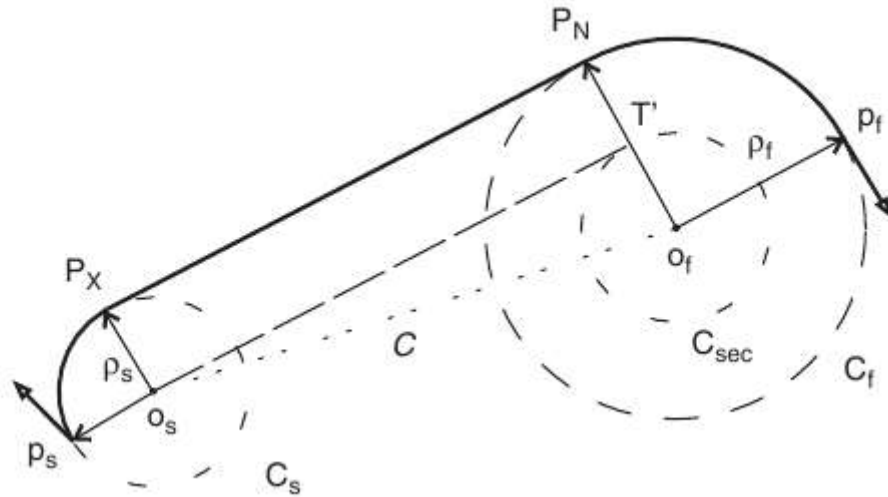


Рисунок 2.4 – Dubins шляхи із зовнішньою дотичною дугою  
Друкується з дозволу з ASME

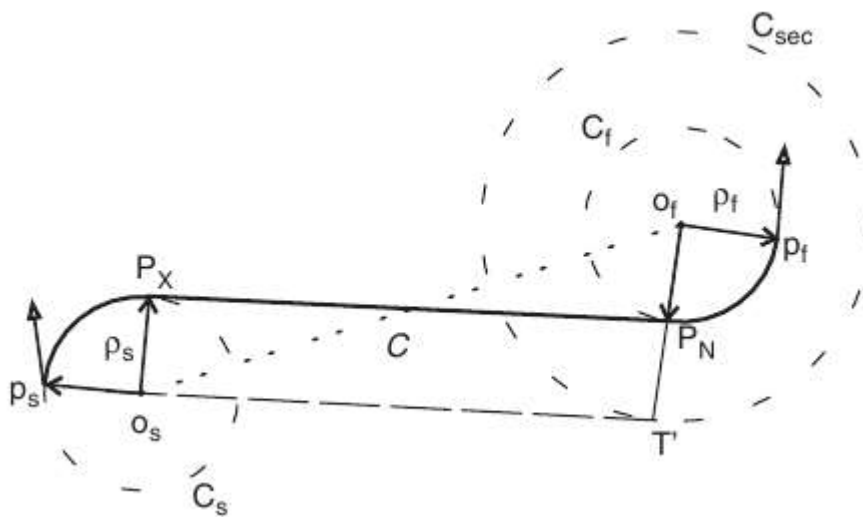


Рисунок 2.5 – Dubins шлях з внутрішньою дотичною дугою

Тут х 'S' це параметри запуску і х 'F' це кінцеві параметри.

Конструкція шляху Dubins математично може бути представлена

$$P_s(x_s, y_s, \varphi_s) \xrightarrow{r(q)} P_f(x_f, y_f, \varphi_f), |K(t)| \leq K_{max} \quad (2.1)$$

Рішення рівняння (2.1) проводить набір Dubins шляхів.

Так як, Dubins шлях складається з шляху з двох дуг кіл і прямолінійного шляху, то довжина шляху Dubins є сумою їх індивідуальних довжини, тобто

$$S_{Dubins} = S_{arc,shart} + S_{tangent} + S_{arc,finish} \quad (2.2)$$

У разі конструкції з використанням аналітичної геометрії, маємо

$$S_{Dubins} = \frac{1}{K_s} \phi_s + s_t + \frac{1}{K_f} \phi_f = f(K_s, K_f) \quad (2.3)$$

де  $S_{Dubins}$  – це довжина шляху Dubins,  $\phi_s$  і  $\phi_f$  відповідно є початкові і кінцеві кути дуги отримані з таблиць 2.1 і 2.2.

З рівняння (2.3) видно, що довжина шляху є просто функцією від початкового радіусу. Тим не менш, ця функція не просте рівняння і, варіюючи значення кривизни, довжину дотичної дуги можна варіювати.

Таблиця 2.1 – Розрахунок дотичного в'їзду і виїзду балів за зовнішню межу

	Початок повороту	Кінець повороту
$\Phi_{right}$	$\alpha + \beta + \pi/2$	$\alpha + \beta + \pi/2$
$\Phi_{left}$	$\beta - \alpha + 3\pi/2$	$\beta - \alpha + 3\pi/2$

Таблиця 2.2 – Розрахунок дотичного в'їзду і виїзду точки для внутрішньої дотичної дуги

	Початок повороту	Кінець повороту
$\Phi_{right}$	$\alpha + \beta$	$\beta - \alpha + \pi$
$\Phi_{left}$	$\beta - \alpha + 2\pi$	$\alpha + \beta + \pi$

Dubins шлях має просту геометрію і, отже, його легко виробляти і реалізувати. Тим не менш, зміни в кривизні траєкторії є переривчастими, коли шлях змінюється від дуги до сегменту прямої лінії і навпаки. Цей розрив може виявитися DIF культом для цілей практичного здійснення, і тому БПЛА повинні слідувати по такому шляху, і він не може миттєво змінити свою поведінку, коли він перетинає кордон сегменту. Зміна кривизни

може бути прямо пов'язана зі зміною поперечного прискорення. Насправді, для БПЛА руху з постійною швидкістю  $V_{MS-1}$ , кривизна шляху слідування і прискорення перпендикулярно до траєкторії (бічне прискорення) задаються

$$a = v^2 \cdot K \quad (2.4)$$

де  $a$  – це бічне прискорення БПЛА і  $K$  – це кривизна шляху.

Тому раптова зміна попиту прискорення відбуватиметься в БПЛА, що перетинає кордони сегмента шляху, помилки будуть виникати в траєкторії після здійснення маневру. Отже, важливо, щоб БЛА слідували шляху, які не є переривчастої кривизни. З принципів диференціальної геометрії, друга похідна кривої по відношенню до параметру шляху це прискорення. (Зверніть увагу, що параметр Шлях може бути будь-який зручний параметр, наприклад, відстань по шляху або інший зручний параметра, як кут повороту для кола). Таким чином, плавний рух вимагає принаймні незникаючих перших двох похідних, геометрично шлях повністю визначається його кривизною, і кривизна пропорційна бічного прискорення для постійної швидкості БПЛА.

У к кривизни (Q) утворення кривизни відбувається за правилом  $(Q) = (x(Q), Y(Q))$ , де Q – це визначений параметр кривої

$$K(q) = \frac{\dot{r} \times \ddot{r}}{|\dot{r}|^3} = \quad (2.5)$$

$$= \frac{\dot{x}(q)\ddot{y}(q) - \dot{y}(q)\ddot{x}(q)}{(\sqrt{\dot{x}^2 + \dot{y}^2})^3} \quad (2.6)$$

де  $\dot{x} = dx/dq$ ,  $\dot{y} = dy/dq$ ,  $\ddot{x} = d^2x/dq^2$  та  $\ddot{y} = d^2y/dq^2$  – це визначення параметру шляху.

З рівняння (2.6), видно, що кривизна функції перших двох похідних кривої визначено, так що шлях повинен бути принаймні двічі безперервно

диференційований; тобто, він повинен мати спадкоємність  $C^2$ . На шляху Dubins, як показано на рис. 2.5 дуги мають постійну кривизну і лінія має нульову кривизну, але межі сегмента мають розрив у кривизні. Тим не менш, крива не має раптової зміни в напрямку, що означає, що шлях має гладку похідну першого, але переривчасту похідну другого порядку. Тому шлях Dubins має спадкоємність  $C^1$ , але не безперервність  $C^2$ . (Позначення  $C^2$  представляє собою безперервність першої похідної і включаючи другу похідну, і  $C^1$  являє собою безперервність включаючи першу похідну.)

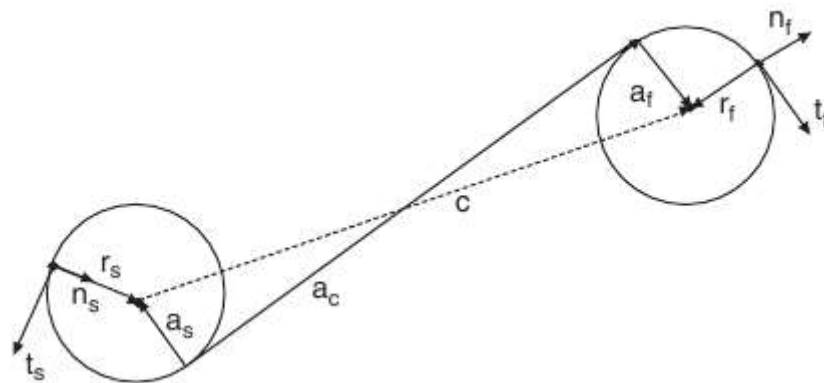


Рисунок 2.6 – Dubins шлях. Друкується з дозволу Американського інституту аеронавтики і астронавтики, і ASME

Шляхи безперервної кривизни можуть бути отримані за допомогою складового шляху, утвореного шляхом об'єднання сегментів, які мають рівні кривизни на їхніх кордонах, або утворивши єдину криву з безперервності  $C^2$ . Першу опцію може бути виконано за рахунок використання клотоїда дуги, які забезпечують безперервність необхідної кривизни через кордони сегментів. Другий варіант зустрічається при використанні компактного, добре визначеного шляху, використовуючи піфагоровські годографи, що впливають на один шлях, а не на колекцію сегментів шляху.

Ці два формати шляху, а саме Dubins і клотоїда, формуються з дугових і прямолінійних сегментів. Кожен сегмент шляху забезпечує будь-який з цих основних рухів. Альтернативний шлях, який забезпечує безперервну кривизну є альтернативним рішенням проблеми планування шляху. Є

широкий вибір поліномів, у яких є мінімальна вимога безперервної кривизни: що виходить із незникаючої другої похідної. Одним з найбільш часто використовуваних поліноміальних форм у плануванні шляху є крива В-сплайну.

Шлях В-сплайн коефіцієнт якого розраховуються з використанням послідовного квадратичного програмування для задоволення обмежень і цілей, детально викладається в Lian і Мюррей (2002). Тим не менш, кривизни обмеження в роботі не розглянуто і, як кривизна будь-якого шляху, шлях В-сплайн повинен відповідати обмеженням БЛА.

Піфагора годографа (РН), були вперше введені Farouki і Sakkalis (1990). Всі криві РН – це визначені поліноміальні криві, які мають годографи, які задовольняють стан Піфагора. Годографи дотичні криві, похідні першої кривої і, отже, прирівнюються з векторами швидкості. Умова Піфагора полягає в тому, що сума квадратів сторін прямокутного трикутника дорівнює квадрату його гіпотенузи.

У тимчасовій області, годографом називається вектор швидкості, який завжди паралельно дотичний до траєкторії. Тим не менш, висновок шляху РН виник через визначення довжини траєкторії. Довжина  $h$  шляху  $r(q)$  параметризовані з параметром  $q$

$$h(q) = \int_{q_1}^{q_2} \|\dot{r}(q)\| dq = \int_{q_1}^{q_2} \sqrt{\dot{x}(q)^2 + \dot{y}(q)^2} dq \quad (2.7)$$

де  $q \in [q_1, q_2]$ , та  $\dot{x}(q) = dx/dq$  та  $\dot{y}(q) = dy/dq$  – годографи

За для розрахунку довжини шляху потрібно рішення інтеграла у рівнянні (2.7). Наявність квадратного кореня в рівнянні не може привести до замкнутої форми розчину в цілому, і розчин повинен бути обчислений шляхом ітерації, і пов'язані з ним проблеми точності та обчислювального навантаження. Краще рішення рівняння довжини шляху потрібно, щоб усунути цю проблему. За допомогою спеціальної поліноміальної форми, Піфагора годограф має замкнуту форму рішення. Це досягається шляхом

зазначення, що термін всередині квадратного кореня з рівняння (2.7) являє собою суму квадратів годографом. Якщо можливо представити член в квадратний корінь, як ідеальний квадрат, скажімо,  $\sigma(Q)^2$ , то рішення довжини шляху буде інтеграл поліноміального рівняння  $\sigma(T)$  який прийме форму

$$\sigma(q)^2 = \dot{x}(q)^2 + \dot{y}(q)^2 \quad (2.8)$$

та

$$h(q) = \int_{q_1}^{q_2} |\sigma(q)| dq \quad (2.9)$$

Це еквівалентно до задоволення теореми Піфагора, приймаючи багаточлени  $\sigma(Q)$  як гіпотенузи, і з  $x(Q)$  у  $(Q)$  в якості двох інших сторін. Поліном кривої, годографи відповідають умові рівняння (2.8) називається Піфагора годографа (РН). Шлях РН призначений вибравши відповідні поліноми  $u(q)$ ,  $v(q)$  і  $w(q)$  побудувати годографи  $X(Q)$  і  $Y(Q)$ , так що вони відповідають умовам в рівнянні (2.8). Функції, які відповідають вимогам рН задаються

$$\dot{x}(q) = w(q)[u(q)^2 - v(q)^2] \quad (2.10)$$

$$\dot{y}(q) = 2w(q)u(q)v(q) \quad (2.11)$$

з яких випливає, що

$$\sqrt{\dot{x}(q)^2 + \dot{y}(q)^2} = \omega(q)[u(q)^2 + v(q)^2] = |\sigma(q)| \quad (2.12)$$

де  $u(q)$  та  $v(q)$  – відносно прості багаточлени ступеня 2  $w(q) = 1$ .

Це призводить до поліному  $\sigma(q)$  ступеня  $(n - 1)$ . Тепер, завдання зводиться до передачі коефіцієнтів поліномів  $u(q)$ ,  $v(q)$  і  $w(q)$ . Перевагою даного формату є не тільки те, що він виробляє замкнуту форму рівняння (2.7), але також, що це призводить до рівномірного розподілу контрольних точок на шляху. Іншими словами, є рівне збільшення довжини шляху при рівному збільшенню параметра  $q$ .

Параметрична швидкість  $Q$  і кривизни ( $Q$ ) кривої РН можуть бути раціональними і задаються

$$\dot{h}(q) = |\sigma(q)| \quad (2.13)$$

$$K = \frac{2[u(q)\dot{v}(q) - \dot{u}(q)v(q)]}{\omega(q)[u(q)^2 + v(q)^2]} \quad (2.14)$$

$$r_d(q) = r(q) \pm dn(q) \quad (2.15)$$

де  $\dot{h}(q) = dh/dq$  і  $n(q)$  це одинична нормаль до кривої  $r(q)$ . В якості додаткового бонусу, зміщення кривої на відстань  $D$  від шляху РН є також шляхом РН. Це буде корисно при розміщенні безпечної відстані навколо шляху до випробування для запобігання зіткнень.

З рівняння (2.13), параметрична швидкість кривої РН це проблема багаточлена, з кривизною, заданою рівнянням (2.14). Зсув кривої в рівнянні (2.15) може бути використаний області безпеки, що визначається або датчик діапазон невизначеності по шляху. Рисунок 2.6 показує порівняльну візуалізацію гладкого шляху РН та шляхів Dubins з тією ж максимально пов'язаною кривизною. Як видно з цього рисунка, безперервність кривизни траєкторії РН досягається зі збільшенням довжини шляху.

Довжина шляху РН повинна бути більше, ніж у траєкторії Dubins, а шлях Dubins найкоротший шлях між будь-якими двома позначками. Безпечна відстань всього шляху рН визначена за два зміщення шляху, показана пунктирними колами. Мінімальний багаточлен порядку, який показує поведінку рН, і називається кубічною РН. Тим не менш, нижчий порядок шляху РН, який має точку перегину є п'ятим колом (Farouki і Sakkalis 1990), називаються квінтіка РН. Наявність у відображенні точки шлях, дозволяє мати більше гнучкості, так що шляхом можна легко маніпулювати. Таким чином, квінтіка крива РН використовується для планування шляху. Відтепер шлях РН позначає п'яту ступінь кривизни.



У початковій і кінцевій позиції відповідно є  $(X_S, Y_S)$  і  $(X_F, Y_F)$ , а відповідні напрями (дотичні напрями) є  $\phi_S$  і  $\phi_F$ . Ці змінні визначаємо як початкові і кінцеві позиції і граничні значення для шляху тіл.

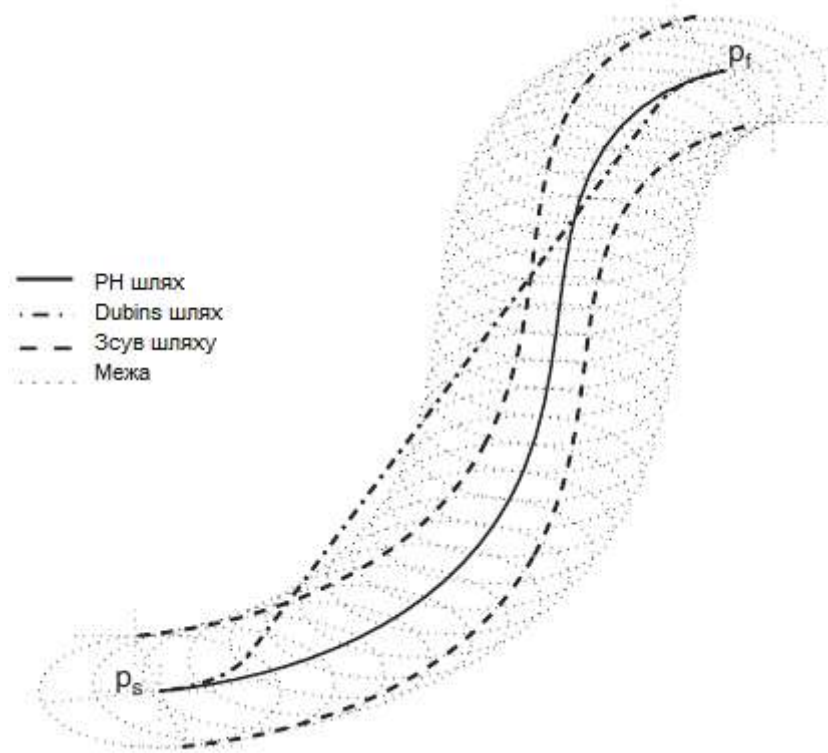


Рисунок 2.7 – Порівняння шляху Dubins з Піфагоровим шляхом годографа

Шлях РН представлено в Безьє вигляді чисельної стабільності. Загальне рівняння для  $n$ -го порядку полінома Безьє має вигляд

$$r(q) = \sum_{k=0}^n b_k \binom{n}{k} q^k (1-q)^{n-k}, \quad q \in [0, 1] \quad (2.16)$$

де  $b_k = (x_k, y_k)$ ,  $k = 1, \dots, n$ , це контрольні точки,

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (2.17)$$

і  $r(q)|_q=0 = 0$  і  $r(q)|_q=1 = 1$ , відповідно, являють собою початкову і кінцеву точки шляху. Зверніть увагу, що параметр Шлях  $Q$  унормований на одиницю.

Тепер,  $r$ -на похідна  $n$ -го порядку кривої Безьє має вигляд

$$\frac{d^r r(q)}{dq^r} = \frac{n!}{(n-r)!} \sum_{j=0}^{n-r} \Delta^r b_j \binom{n-r}{j} q^j (1-q)^{(n-r-j)} \quad (2.18)$$

$$\Delta^r b_i = \sum_{j=0}^r \binom{r}{j} (-1)^{r-j} b_{i+j}$$

де

$$\Delta b_j = b_{j+1} - b_j \quad \Delta^0 b_j = b_j$$

Для квінтіку шляху,  $n = 5$  і рівняння (2.16) стає

$$r(q) = \sum_{k=0}^5 q_k \binom{5}{k} q^k (1-q)^{(5-k)} \quad (2.19)$$

або

$$r(q) = b_0(1-q)^5 + 5b_1q(1-q)^4 + 10b_2q^2(1-q)^3 + 10b_3q^3(1-q)^2 + 5b_4q^4(1-q) + b_5q^5 \quad (2.20)$$

З рівняння (2.18), перша похідна шляху  $r(q)$  має вигляд

$$\frac{dr(q)}{dq} = 5 \sum_{j=0}^4 \binom{4}{j} \Delta^1 b_j q^j (1-q)^{4-j} \quad (2.21)$$

або

$$\frac{dr(q)}{dq} = 5(b_1 - b_0)(1-q)^4 + 20(b_2 - b_1)q(1-q)^3 + 30(b_3 - b_2)q^2(1-q)^2 + 20(b_4 - b_3)q^3(1-q) + 5(b_5 - b_4)q^4 \quad (2.22)$$

Тепер ці рівняння можуть бути використані для інтерполяції між двома позиціями і визначено форму сполучної кривої. Так як положення і напрямки на початковій і кінцевій точках відомі, використовуємо інтерполяції першого порядку.

Підставляючи координати положення при  $T = 0$  і  $t = 1$  в рівняння (2.20) і використовуючи першу похідну шляху по рівнянню (2.22), контрольні точки  $b_0$ ,  $b_1$ ,  $b_5$  і  $b_4$  розраховується як

$$b_0 = (x_s, x_s),$$

$$b_5 = (x_f, x_f),$$

$$d_0 = (\cos(\varphi_s), \sin(\varphi_s))$$

$$\begin{aligned}
 d_5 &= (\cos(\varphi_f), \sin(\varphi_f)) \\
 b_1 &= b_0 + 1/5 d_0 \\
 b_4 &= b_5 + 1/5 d_5
 \end{aligned}
 \tag{2.23}$$

Таким чином, контрольні точки ( $b_0, b_1, b_4, b_5$ ) в рівнянні (2.23) фіксовані до початку. Тепер завдання зводиться до відбору двох контрольних точок  $b_2$  і  $b_3$  так, щоб рівняння (2.20) задовольняло умові РН в рівнянні (2.8). Це призводить до чотирьох рішень (Farouki і Neff 1995). Мінімальне рішення кривої енергії від Farouki (1996), яке має плавну зміну кривизни, використовується для планування шляху.

У попередньому підрозділі було розроблено теорію та практичне застосування планування маршруту в двох вимірах. На практиці, маневрування відбувається в трьох вимірах. Тривимірне маневрування необхідне, наприклад, під час зльоту, посадки, зміни висоти Безпілотного літального апарату (далі скорочено – UAV, англ.), або коли він повинен літати над перешкодою, а не навколо.

Використання двовимірного планування маршруту часто зустрічається у літературі. Це може бути пов'язане з невеликою кількістю досліджень планування маршруту наземних роботів. Тим не менш, при роботі з літальними апаратами, до площини руху необхідно додати висоту у розрахунках маневрів в космосі. Існує лише кілька посилань на роботи, виконані в трьох вимірах для планування маршруту. Досить широко планування руху на основі вибірки в багатовимірному просторі вивчалось у NSU (Гумбольдтському державному університеті) (1997 р.) та Л. Кавракі (1996 р.). Вивчення тривимірних маневрів було продовжене Дубінім у Шанмугавелі та іншими у 2006 році. Аналогічний підхід можна побачити в теорії Пачікара та ін. у 2009 році.

З точки зору планування траєкторії, початок у плануванні шляху залишається таким самим, тобто, параметри однакові у початковій та

кінечній точках. Фактично є три просторові координати (x, y, г), і три кутові орієнтації, (крену, тангажу, нишпорення). У цій главі деталізовано алгоритми для виробництва траєкторії польоту 3D. Дослідження проведене на прикладі двох методів, за аналогією з 2D алгоритмами з попереднього підрозділу, а саме: (I) траєкторія Дубіна і (II) просторова теорема Піфагора щодо годографа(далі - РН).

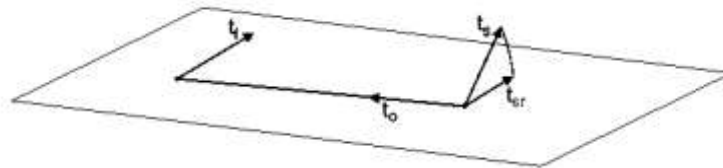


Рисунок 2.8 – Тривимірні умови маневру Dubins

Тривимірний шлях польоту  $r(q)$  можна отримати за допомогою рішення наступного рівняння:

$$P_s(x_s, y_s, z_s, \theta_s, \psi_s) \xrightarrow{r(q)} P_f(x_f, y_f, z_f, \theta_f, \psi_f) \quad (2.24)$$

$$|k(t)| < k_{max}, |\tau(t)| < \tau_{max}$$

де  $\tau(t)$  кривизна,  $\chi(t)$  – кручення траєкторії

Одна з класичних траєкторій, що використовується для маневру літального апарату від однієї висоти до іншої є перетином круглої спіралі, яка проектується на площину X-Y у вигляді кола. Траєкторію можна представити як намотування на поверхню вертикального циліндра. Важливою властивістю цієї кривої являється те, що і кривизна і кручення постійні, з відношенням до двох дає радіус циліндра і виток спіралі. У порівнянні з Траєкторією Dubins та РН, довжина траєкторії спіралі буде більше, ніж будь-яка інша та більш сувора за формою траєкторії.

Корисніший набір кривої можна отримати з узагальнення Дубінського та Піфагорського годографу шляху з 2D в 3D. Якщо два дотичних вектора  $t_s$  і  $t_f$  з'єднані лінією вектора  $t_o$ , як показано на рисунку 2.8, то 2D маневр може

бути виконаний за умови, що всі вектори лежать в одній площині. Рисунок показує, що два дотичні вектори знаходяться в різних площинах, а початок дотичного вектора  $t_s$  не лежить в площині, яка визначається двома векторами  $t_f$  і  $t_0$ .

Один підхід для отримання 3D-траєкторії, спочатку, щоб визначити початковий маневр дуги, що торкається дотичного вектору  $t_s$  в площині, щоб отримати  $t_{sr}$  (Рисунок 2.9). Це буде гарантувати, що цей вектор знаходиться в одній площині з  $t_0$  і  $t_f$ , та траєкторію 2D Dubins можна обчислити за допомогою повернутого вектора  $t_{sr}$  в якості нового вектора початку дотичної. Нормальний вектор  $n_{sr}$  може бути визначений як норма до дотичного вектора, а також лежить в площині, яка визначається  $t_f$  і  $t_0$ . Бінормалі вектору  $b_{sr}$  потім складає праву трійку.

Початкова траєкторія буде складатися з комбінованої кривизни траєкторії щодо вектору бінормалі, з подальшим креном, щоб вибудувати нормальні і бінормалі вектори, для формування остаточної 2D траєкторії Dubins. Тому композиція траєкторії Dubins складається з однієї площини кола траєкторії з визначенням кривизни  $k_s$ , а потім нормальний запуск траєкторії кола з визначенням кривизни  $k_s$ , прямолінійної траєкторії і, нарешті, закінчення траєкторії кола з визначенням кривизни  $k_f$ . Геометрія, яка визначає остаточної траєкторію 2D наведено у попередньому підрозділі. Рисунок 2.8 показує, отриману 3D траєкторію Dubins.

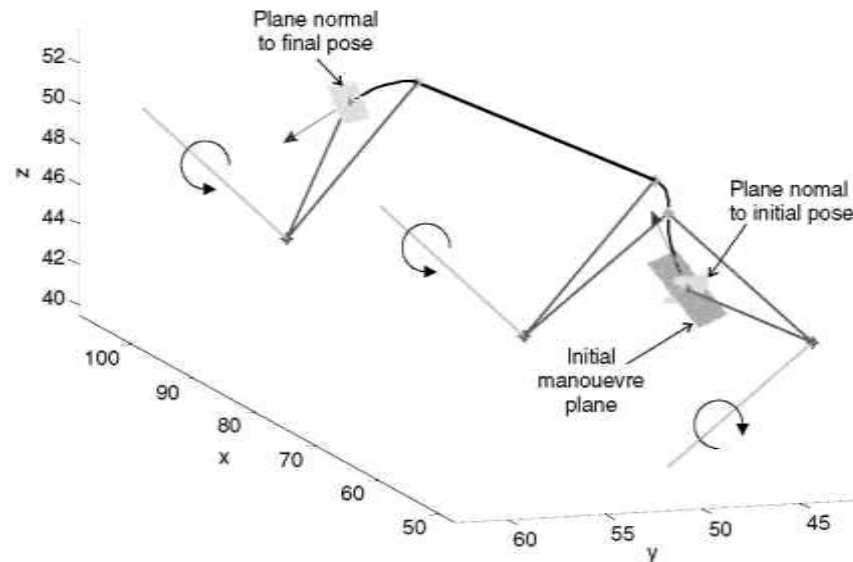


Рисунок 2.9 – Тривимірна траєкторія Dubins в UAV

Більш прямий підхід, для усунення початкового обертання в конфігурації траєкторій в одній площині, це виконання повного 3D маневру з використанням 3D геометричного аналізу. Два підходи буде докладно описано в цій главі. Перше вірне вирішення 3D Dubins полягає у диференційно-геометричних концепціях, інший полягає у використанні 3D просторової траєкторії Піфагора РН, який отримують способом, аналогічним алгоритму 2D з використанням першого порядку ермітової інтерполяції. Отримана траєкторія додатково налаштована на збільшення траєкторії польоту шляхом збільшення довжини граничних дотичних векторів.

Як вже зазначалося, в разі 2D траєкторії Dubins, рама  $(t_s, n_s)$  і  $(t_f, n_f)$  лежать в одній і тій же траєкторії літака. Тобто, бінормалі вектору в обох точках нормальні до площини траєкторії і, таким чином, паралельні один одному. Але для багатьох точок в 3D просторі, це не вірне правило

Знак маневру може бути визначений з урахуванням осьової лінії між двома положеннями. З точки зору кожної позиції, позитивне чи негативне обертання від дотичного вектора до центрального вектора буде визначено знаком кривизни для кожного маневру. Початок маневру і на його фініші літак виходить поворотом про дотичного вектора  $t_s$ . В нас є

$$[t_{ms} \quad n_{ms} \quad b_{ms}] = [t_s \quad n_s \quad b_s]R_s$$

$$[t_{mf} \quad n_{mf} \quad b_{mf}] = [t_f \quad n_f \quad b_f]R_f \quad (2.24)$$

де

$$R_s = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi_s) & -\sin(\varphi_s) \\ 0 & \sin(\varphi_s) & \cos(\varphi_s) \end{pmatrix}$$

$$R_f = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi_f) & -\sin(\varphi_f) \\ 0 & \sin(\varphi_f) & \cos(\varphi_f) \end{pmatrix} \quad (2.25)$$

і де  $\varphi_s$  і  $\varphi_f$  кути повороту для початкових і фінішних маневрів літака.

Радіус-вектори може бути визначено в маневрених літаків, як

$$r_s = [t_{ms} \quad n_{ms} \quad b_{ms}] \begin{pmatrix} 0 \\ \mp 1/k_s \\ 0 \end{pmatrix} \quad (2.26)$$

і, аналогічно,

$$r_f = [t_{mf} \quad n_{mf} \quad b_{mf}] \begin{pmatrix} 0 \\ \mp 1/k_f \\ 0 \end{pmatrix} \quad (2.27)$$

Базисні вектори пов'язані

$$[t_f \quad n_f \quad b_f] = [t_s \quad n_s \quad b_s]R \quad (2.28)$$

де  $R$  є матрицею обертання на яку потрібно змінити вісь набору від початку до фінішу.

Таким чином, ми маємо

$$R = (t_f \quad n_f \quad b_f) \cdot (t_s \quad n_s \quad b_s) \quad (2.29)$$

дає

$$R = \begin{pmatrix} t_f \cdot t_s & t_f \cdot n_s & t_f \cdot b_s \\ n_f \cdot t_s & n_f \cdot n_s & n_f \cdot b_s \\ b_f \cdot t_s & b_f \cdot n_s & b_f \cdot b_s \end{pmatrix} \quad (2.30)$$

З'єднувальні вектори утворюють ортогональний набір векторів. З'єднувальні вектори нормальні до вектора змінного струму, лежать паралельно. Кожен вектор лежить у відповідній площині маневру, які не

збігаються. Внутрішній з'єднувальний вектор змінного струму є загальним для обох маневрів літаків. Таким чином, можна записати у вигляді

$$\begin{aligned} a_c &= a[t_{ms} \quad n_{ms} \quad b_{ms}] \alpha_s \\ &= a[t_{mf} \quad n_{mf} \quad b_{mf}] \alpha_f \end{aligned} \quad (2.31)$$

3

$$\alpha_s = \begin{pmatrix} \alpha_{ts} \\ \alpha_{ns} \\ \alpha_{bs} \end{pmatrix} \quad \text{та} \quad \alpha_f = \begin{pmatrix} \alpha_{tf} \\ \alpha_{nf} \\ \alpha_{bf} \end{pmatrix} \quad (2.32)$$

Кадри Френе для двох площин маневру можуть бути пов'язані з

$$\begin{aligned} [t_f \quad n_f \quad b_f] &= [t_s \quad n_s \quad b_s] R \\ [t_{mf} \quad n_{mf} \quad b_{mf}] &= [t_f \quad n_f \quad b_f] R_f \\ [t_{ms} \quad n_{ms} \quad b_{ms}] &= [t_s \quad n_s \quad b_s] R_s \end{aligned} \quad (2.33)$$

Через

$$\begin{aligned} [t_{ms} \quad n_{ms} \quad b_{ms}] R'_s &= [t_s \quad n_s \quad b_s] \\ [t_{mf} \quad n_{mf} \quad b_{mf}] R'_f &= [t_f \quad n_f \quad b_f] \\ &= [t_s \quad n_s \quad b_s] R \\ &= [t_{ms} \quad n_{ms} \quad b_{ms}] R'_s R \end{aligned} \quad (2.34)$$

та

$$[t_{mf} \quad n_{mf} \quad b_{mf}] = [t_{ms} \quad n_{ms} \quad b_{ms}] R'_s R R_f \quad (2.35)$$

Це означає, що

$$\begin{aligned} \alpha_s &= R'_s R R_f \alpha_f \\ \alpha_f &= R'_f R R_s \alpha_s \end{aligned} \quad (2.36)$$

Радіус-вектори RS і RF також можуть бути описані в осях початку маневру, як

$$r_s = [t_{ms} \quad n_{ms} \quad b_{ms}] \begin{pmatrix} 0 \\ \mp 1/k_s \\ 0 \end{pmatrix}$$



$$r_f = [t_{ms} \quad n_{ms} \quad b_{ms}]R'_sRR_f \begin{pmatrix} 0 \\ \mp 1/k_f \\ 0 \end{pmatrix} \quad (2.37)$$

Тепер, вектори, лежать в маневрених літаків і нормальні до сполучного вектора змінного струму. Вони також можуть бути визначені в осях початку маневру, у вигляді

$$\begin{aligned} a_s &= \frac{\mp 1}{k_s} [t_{ms} \quad n_{ms} \quad b_{ms}] \beta_s \\ a_f &= \frac{\mp 1}{k_f} [t_{mf} \quad n_{mf} \quad b_{mf}] \beta_f \\ &= \frac{\mp 1}{k_f} [t_{ms} \quad n_{ms} \quad b_{ms}] R'_s R R_f \beta_s \end{aligned} \quad (2.38)$$

щоб гарантувати, що вектори з'єднання лежать у площині маневру і нормальні до внутрішнього вектору відбувається підключення до мережі, як

$$\begin{aligned} \beta_s &= \frac{1}{b_s} \begin{pmatrix} -\alpha_{ns} \\ \alpha_{ts} \\ 0 \end{pmatrix}, \quad \beta_f = \frac{1}{b_f} \begin{pmatrix} -\alpha_{nf} \\ \alpha_{tf} \\ 0 \end{pmatrix} \\ b_s &= \sqrt{\alpha_{ns}^2 + \alpha_{ts}^2}, \quad b_f = \sqrt{\alpha_{nf}^2 + \alpha_{tf}^2} \\ \beta_s \alpha_s &= 0, \quad \beta_f \alpha_f = 0 \end{aligned} \quad (2.39)$$

Положення фінішної точки щодо стартової позиції вимірюється в початковій площині осей, як

$$\begin{aligned} p_s - p_s &= [t_s \quad n_s \quad b_s] p \\ &= [t_{ms} \quad n_{ms} \quad b_{ms}] R'_s p \\ p_m &= R'_s p \\ p &= \begin{pmatrix} p_t \\ p_n \\ p_b \end{pmatrix} \end{aligned} \quad (2.40)$$

Потім, векторна сума для вектора позиції задається

$$p_m = -r_s + a_s + a_c - a_f + r_f$$

$$p_m + r_s - r_f = a_s + a_c - a_f \quad (2.41)$$

Підставляючи вектори, отримаємо

$$R'_s p + r_s - R'_s R R'_f r_f = a_s + a_c - R'_s R R'_f a_f \quad (2.42)$$

Перепишучи це в початковій площині осей дає

$$p + R_s r_s - R R'_f r_f = R_s a_s + R_s a_c - R R'_f a_f \quad (2.43)$$

Шлях Dubins в 3D формує композитний шлях трьох або чотирьох сегментів. Довжина шляху Dubins це сума довжини цих сегментів яка задається як

$$h_{Dubins} = h_i + h_s + h_t + h_f = \frac{\alpha_i}{k_s} + \frac{\alpha_s}{k_s} + a + \frac{\alpha_f}{k_f} \quad (2.44)$$

де  $h$  – це довжина шляху,  $S$ ,  $T$  і  $F$ , відповідно, початкова дуга, початок дуги, пряма довжини  $a$  і дугових сегментів. Також  $\alpha$  і  $k$  кути дуги та асоційовані з ними кривизни. Для більш прямого диференційно-геометричного рішення, цей термін відсутній.

Піфагора годограф багаточлен кривої введений Farouki (Farouki і Sakkalis 1990, 1994; Farouki та ін 2002). Як і у випадку 2D, крива PH п'ятого порядку використовується, так як це крива нижчого порядку, яка має точки перегину, які забезпечують достатню, гнучкість у формі шляху, щоб шляху підходив для планування (Farouki і Sakkalis 1990). Шлях PH забезпечує в замкнутій формі поліномів для довжини шляху його кривизни і зміщення кривої. Підставляючи відповідний багаточлен  $\sigma(p)$ , у змінну  $\sigma(q)^2 = \dot{x}(q)^2 + \dot{y}(q)^2 + \dot{z}(q)^2$  рівняння (2.45) виробляє довжини шляху  $h(Q)$  і швидкості  $\dot{h}(Q)$ , які зводяться до інтегралу поліноміальних  $\sigma(Q)$  і самого полінома, відповідно. Канал трубки по всьому шляху також раціональний, він використовується для безпеки навколо кожного БЛА.

Довжина шляху  $h(Q)$  кривизни  $r(Q) = \{x(q), y(q), z(q)\}$  задається

$$h(q) = \int_{q_1}^{q_2} \sqrt{\dot{x}(q)^2 + \dot{y}(q)^2 + \dot{z}(q)^2} dq \quad q \in [q_1, q_2] \quad (2.45)$$

Термін всередині квадратного кореня в рівнянні (2.45) являє собою суму квадратів з годографів. Якщо  $x(q)$ ,  $y(q)$ ,  $z(q)$  поліноміальні функції шляху змінної  $Q$ , є можливість зробити з цього терміну ідеальний квадрат, а довжина шляху буде просто невід'ємною частиною багаточлена  $\sigma(Q)$ , аналогічним чином як проектування у 2D. Так

$$\sigma(q)^2 = \dot{x}(q)^2 + \dot{y}(q)^2 + \dot{z}(q)^2 \quad (2.46)$$

та

$$h(q) = \int_{q_1}^{q_2} |\sigma(q)| dq \quad (2.47)$$

Для будь-якого багаточлена кривої, якщо його годографи вбирають стан Піфагора, крива називається Піфагорова крива годографа. Тепер, для полінома кривої, тобто, коли  $x(Q)$ ,  $y(Q)$  та  $z(Q)$  є поліномами, безпечний шлях РН розроблений щодо вибору відповідних поліномів для годографів,  $X(Q)$ ,  $Y(Q)$  і  $Z(Q)$ . Основні переваги цього формулювання:

- (I) розрахунок довжини шляху без будь-яких наближень;
- (II), рівний приріст відстані вздовж кривої за рівне збільшення параметра  $Q$ ;
- (III) раціональна параметрична швидкість;
- (IV) раціональні власні властивості (кривизна, крутіння і поверхневий канал).

Рівняння (2.48) є багаточленом розробленим щодо інтерполяції вільних векторів на кордоні. Вільні вектори мають позиції  $(X, Y, Z)$  і напрям  $(\theta, \psi)$  в просторі. Крива інтерполяції двох таких векторів називається інтерполяцією Ерміта. Отримана крива матиме спадкоємність дотичну, але не спадкоємність кривизни. Для застосування БЛА, важливо, щоб безперервність кривизни була пропорційна бічного прискорення в БЛА. Таким чином, ми повинні мати контрольовану кривизну на кордонах кривих інтерполяції, а також накласти обмеження максимальної кривизни.

$$Q(q) = \sum_{i=0}^2 Q_i \binom{2}{i} q^i (1-q)^{2-i}, \quad q \in [0,1] \quad (2.48)$$

Це дозволить забезпечити рівний шлях для БПЛА. Тепер, крива РН представлена в п'ятому порядку поліноміальної форми як

$$r(q) = \sum_{k=0}^5 b_k (1-q)^{(5-k)} q^k, \quad q \in [0,1] \quad (2.49)$$

де  $b_k = (x_k, y_k, z_k)$  – контрольні точки, вершини яких визначає управління багатокутника або багатокутника Безьє з  $K = 0, \dots, 5$ . початковими і кінцевими конфігураціями є PS  $(X_S, Y_S, Z_S, \theta_S, \psi_S)$  і P  $(X_F, Y_F, Z_F, \theta_F, \psi_F)$ , відповідно. Чотири контрольні точки полігонів Безьє розраховується шляхом першого порядку ермітової інтерполяції наступним чином:

$$\begin{aligned} b_0 &= (x_s, y_s, z_s), \\ b_5 &= (x_f, y_f, z_f), \\ d_0 &= m_0 [\cos(\theta_s) \cos(\psi_s), \cos(\theta_s) \sin(\psi_s), \sin(\theta_s)] \\ d_5 &= m_5 [\cos(\theta_f) \cos(\psi_f), \cos(\theta_f) \sin(\psi_f), \sin(\theta_f)] \\ b_1 &= b_0 + 1/5 d_0 \\ b_4 &= b_5 + 1/5 d_5 \end{aligned} \quad (2.50)$$

При орієнтації використано сферичні координати. Позитивні константи  $m_0$  і  $m_5$  відіграють вирішальну роль у складанні кривої інтерполяції.

Проблема обходу перешкод тісно пов'язана з плануванням шляху, тому що наявність перешкод, як правило, призводить до перепланування шляхів.

Підтвердження принципу конфігурації простору показано в Perez (1983) для робототехніки це зменшує розмір роботів точкових мас і компенсує це зниження за рахунок збільшення чисельності перешкод. Бази даних являють собою графіки, і дерева, що утворюють мережу вузлів і ребер ізолюючих перешкод. Алгоритм пошуку використовується для підключення початкової та бездротової точки через вузли мережі. Гібридний підхід, в якому маршрут планується для відомих перешкод деформується для обробки

невідомих перешкод використовується в Lamiraux та ін. (2004). Подібний метод деформації запланованого шляху, змінюючи його кривизну проявляється в Shanmugavel та ін. (2005). Тим не менш, реактивний метод, відповідаючи на датчик зворотного зв'язку, є істотним для обробки швидко рухомих динамічних перешкод.

Методи обходу перешкод були вивчені з використанням потенційних полів, де сили тяжіння і відштовхування, відповідно, породжені цілі і перешкоди (Kim і Khosla 1992). Потенціал поля використовується, щоб уникнути перешкод в галузі планування шляху декількох безпілотних літальних апаратів по FL потенційного поля.

Аналітичний і дискретний підхід оптимізації використовується в Zabaranin та ін. (2002) для оптимального покоління шляху ризиками в 2D просторі, з обмеженням по довжині шляху. Змішання цілочисельного лінійного програмування (MILP) використовується в Richards і як (2002) для запобігання зіткнень. В результаті шлях від обох цих підходів не бере до уваги обмеження кривизни, і так додаткова робота повинна бути зроблена, щоб справити необхідний шлях. Запобігання зіткнень використовується для обходу перешкод декількох мобільних роботів у Fujimori та ін. (2002), де два роботи співпрацювали в рамках спеціальної області, в той час як інші чекають за межами області. Однак те, що є допустимим для мобільних роботів, де робот може зупинитися і виконувати розворот, не представляється можливим для застосування БЛА, де БЛА не може зупинитися і де вони мають обмеження на кривизну, накладену на їхньому шляху.

Шлях планування з обходом перешкод в 3D більш складніший. Ускладнення виникає, тому що є в нескінченно багато напрямків для маневрів як для БПЛА так і для перешкод. Крім того, інші обмеження, особливо найкоротша довжина шляху, є більш складним в 3D-просторі.

Дерево октантів уявлення 3D простору було запропонована в Kitamura та ін. (1995, 1996), де простір розділено на доступні для пошуку регіони або

клітин і зіткнення вільного пробігу, отримані шляхом застосування потенційних полями для кожного осередку Дерева октантів. Проблема обходу перешкод була вивчена в різних додатках: щоб звести до мінімуму ризик виявлення літаків – радари, датчики і зенітно-ракетні комплекси, щоб звести до мінімуму ризик виявлення підводного човна – датчики (Washburn, 1990), або максимізувати ймовірність виявлення цілі – пошукачі. Іншою областю застосування в цьому науково-дослідному полі є використання безпілотних літальних апаратів в цивільному повітряному просторі, де для запобігання зіткнень алгоритм еквівалентний тому, що є в комерційних літаків. Існуюча система використовує Трафік оповіщення та систему попередження зіткнень (TCAS), яка випускає рекомендації з вирішення загрози (РАН) на основі позиційних даних і висотою від транспондерів на борту повітряного судна і від наземних датчиків, якщо такі є.

Аспекти безпеки використання безпілотних літальних апаратів в цивільному повітряному просторі обговорюється в Dowek та ін. (2001) і Цейтлін і Маклафлін (2007). Ці активні дослідження області і повинні бути розроблені до позиції, де такі алгоритми є надійними і приймаються до БЛА.

Для того, щоб маневрувати в навколишньому середовищі, яке повно перешкод, БПЛА повинен бути здатний обчислити траєкторії, які будуть плавати навколо цих перешкод і приходити до остаточної пози та положення.

Отже, там є вимога, щоб обчислити композитні траєкторії Dubins, які включають маневри ухилення.

Розглянемо простий випадок єдиної перешкоди, що перетинає прямолінійний сегмент стандартної траєкторії Dubins, як показано на рисунку 2.9. Траєкторія Dubins зараз складається з п'яти сегментів, з початкових, проміжних і кінцевих маневрів дуги, разом з двома прямими відрізками. У початковій і кінцевій дузі маневри модифіковано, щоб відрізки були дотичними до перешкоди кола. Тому два варіанта уникнення траєкторії повинні бути розраховані для кожної перешкоди. Два варіанти відбуваються

в цьому випадку: перешкода перетинає прямолінійний сегмент Dubins траєкторії, як показано на рисунку 2.9.

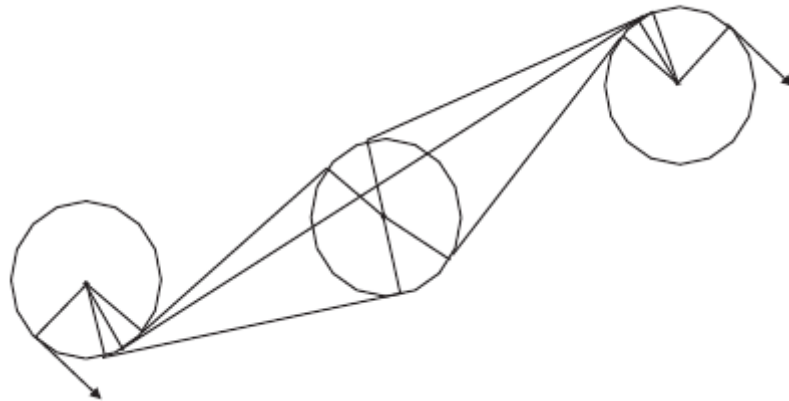


Рисунок 2.10 – Одинична перешкода

Якщо БЛА виявляє перешкоду за допомогою датчика, розташованого на борту, БЛА повинен повторно спланувати шлях або шляхом зміни кривизни між двома точками розглянутого випадку або перепланувати шлях з використанням проміжної точки. Перешкоди проходять перевірку визначення, якщо безпека перешкоди перетинає безпеку БПЛА. Якщо перетин не порожній, то перепланування не потрібно. Це може бути зроблено або шляхом збільшення кривизни траєкторії, або шляхом створення проміжної точки і вироблено новий шлях, який включає в себе цю нову колійну точку.

Якщо припустити, що  $O_{obs}$  є перешкодою кола і безпеки БПЛА  $O_{safe}$ , умовою для уникнення зіткнення буде

$$O_{obs} \cap O_{safe} = \emptyset. \quad (2.51)$$

Схема для перевантаженого середовища показана на рис. 2.10. Перешкоди моделюються у вигляді багатокутників. Області, які передбачається стаціонарно та їх позиції відомі. Планування шляху з

декількох безпілотних літальних апаратів через послідовність поз можна записати у вигляді

$$P_{s,i,j-1}(x_{s,i,j-1}, y_{s,i,j-1}, \theta_{s,i,j-1}) \xrightarrow{r_{i,j-1}(q)} P_{f,i,j}(x_{f,i,j}, y_{f,i,j}, \theta_{f,i,j})$$

$$i = 1, \dots, n_{UAV}, j = 2, \dots, n_p, |k_i(q)| < k_{max}, \Pi_{safe}, \Pi_{lenght} \quad (2.52)$$

де  $n_{UAV}$  – це число безпілотних літальних апаратів,  $n_{UAV}$  – кількість позицій (в тому числі час початку і кінця). Прийнемо два БЛА для реалізації алгоритму. Обидва БЛА є однорідними за своїми фізичним можливостями, і, отже, вони обидва рухаються з тією ж швидкістю і мають обмеження кривизни.

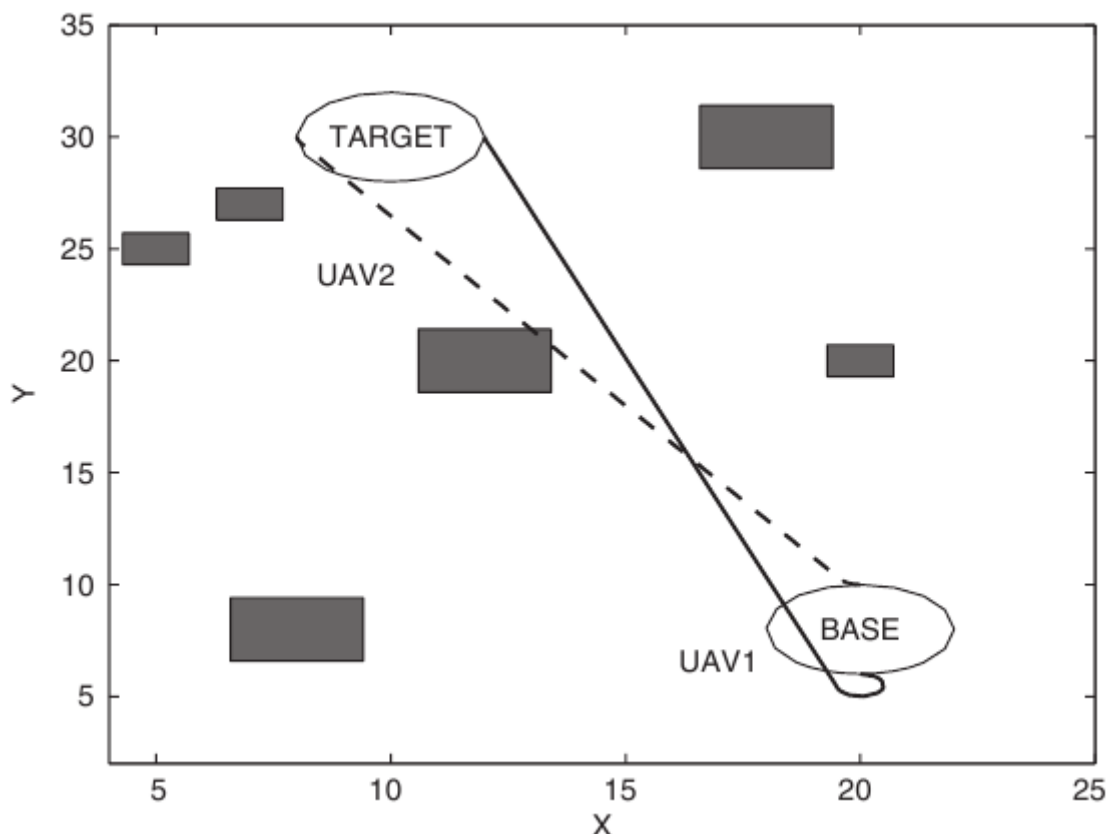


Рисунок 2.11 – Dubins шляхи двох безпілотних літальних апаратів у навколишньому середовищі з перешкодами



Рисунок 2.10 показує траєкторії двох безпілотних літальних апаратів у навколишньому середовищі з перешкодами. Шлях польоту другого безпілотника UAV2 перетинає перешкоду і шляху польоту першого безпілотника UAV1. Кривизна дуги шляху Dubins різноманітна щоб уникнути регіон загрози. Рисунок 2.11 показує, новий безпечний шлях після збільшення кривизни траєкторії UAV2. Рисунок 2.12 показує рішення тієї ж задачі за допомогою проміжної точки.

Після того, як перешкоду перетину було виявлено, проміжні точки генерується.

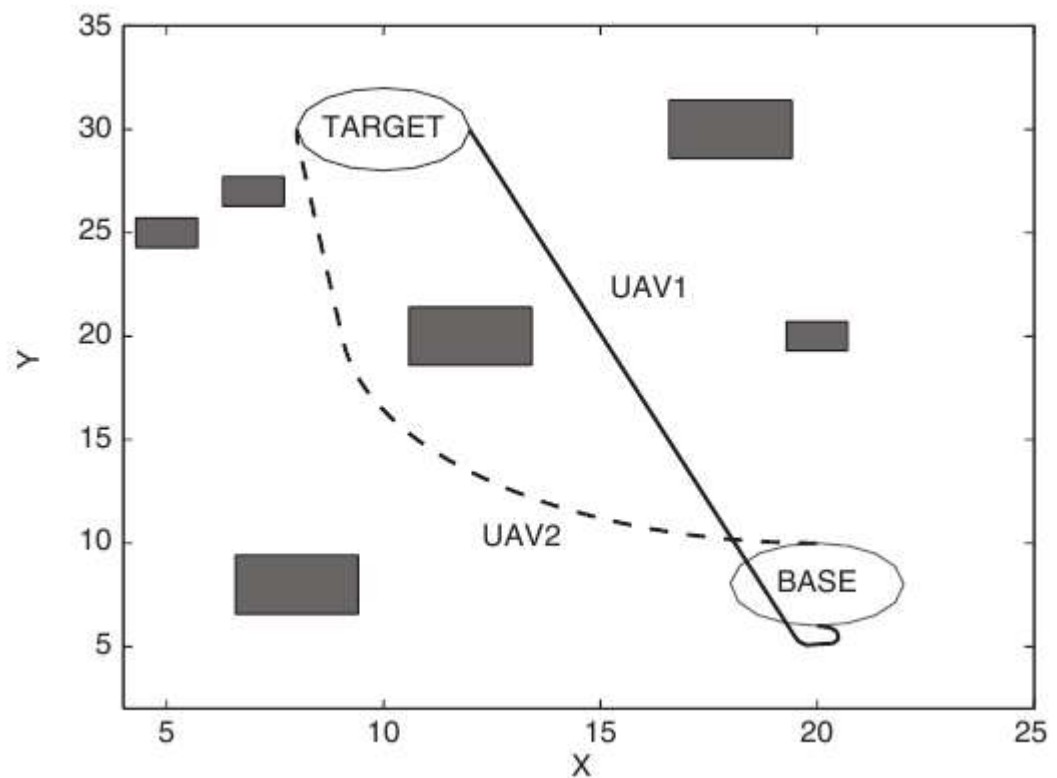


Рисунок 2.12 – Перепланування шляху Dubins UAV2 з регулюванням кривизни

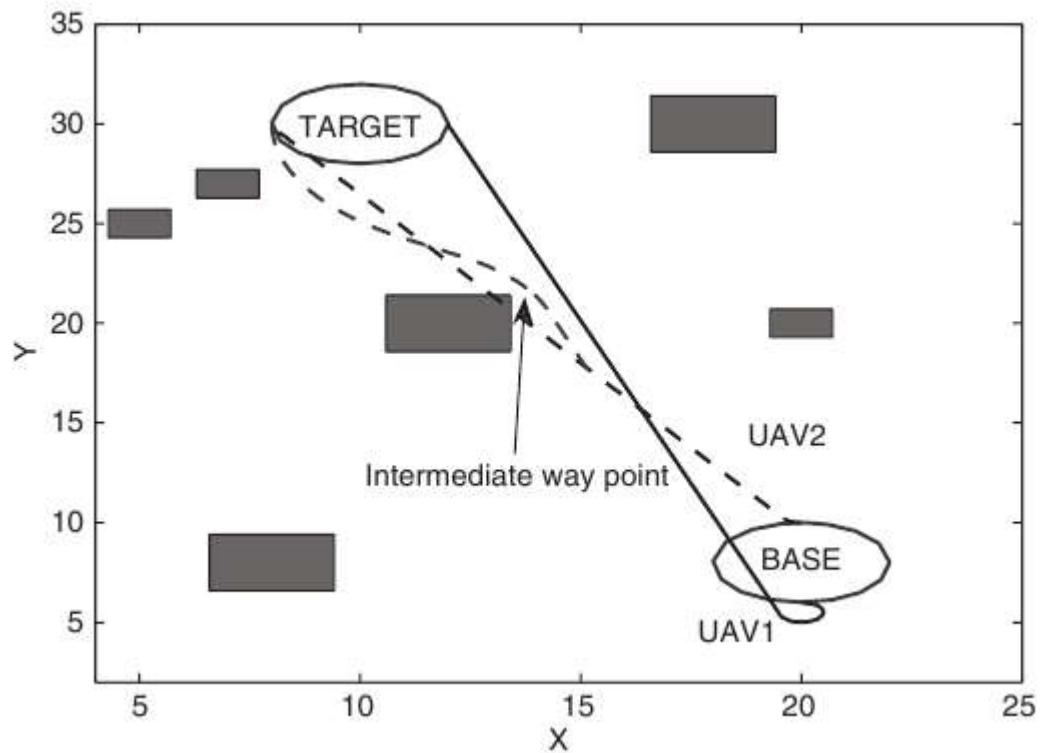
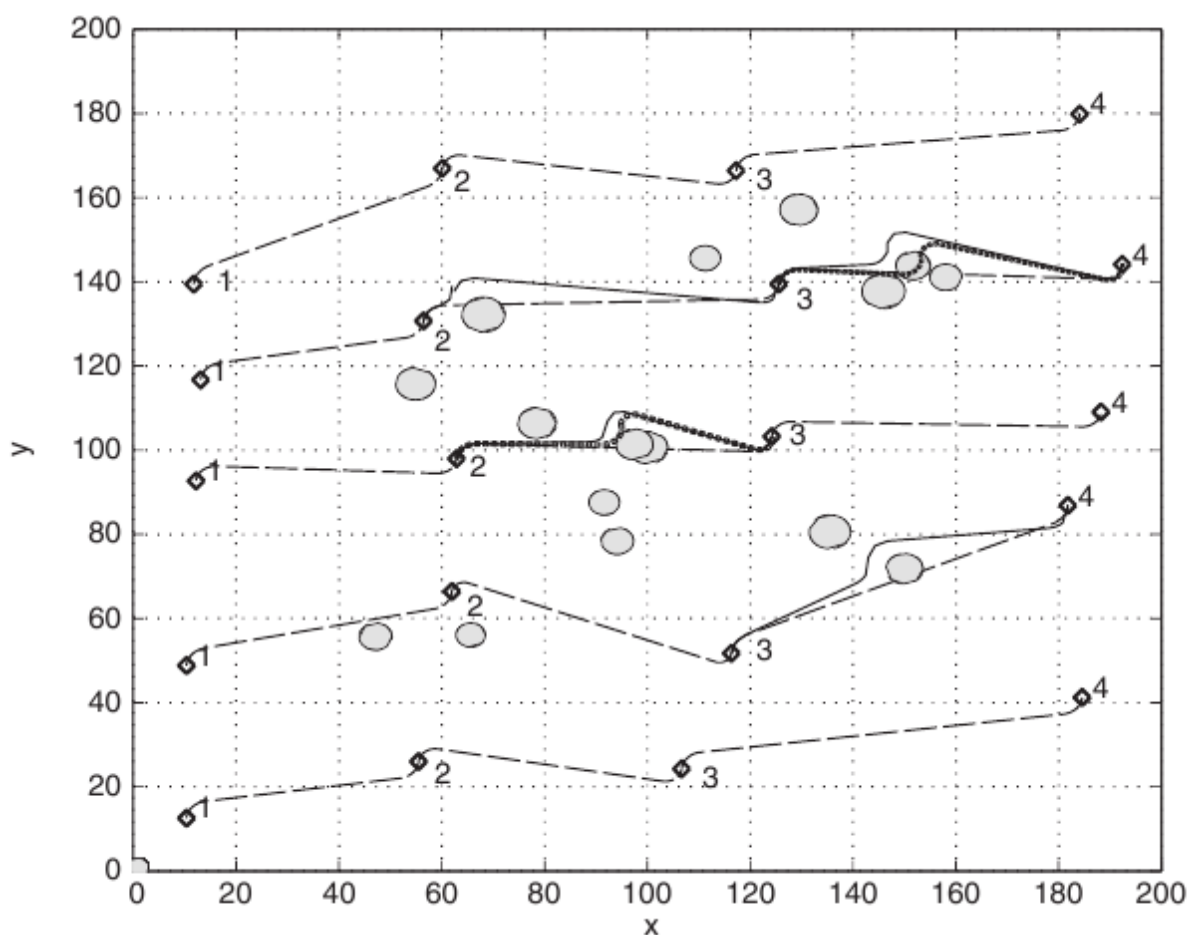


Рисунок 2.13 – Перепланування шляху Dubins UAV2 за допомогою проміжної точки

Перешкоди перетину обробляються шляхом створення проміжних точок та / або пози. Перетин шляхів визначається інтерактивно для клотоїди дугових сегментів, а перетини ліній з еліпсів можуть бути виявлені за допомогою простої геометрії.

Проміжний шлях проводиться для кожного перехрестя в межах зони безпеки перешкоди. Після цього, шляхи переплановуються, щоб пройти через нові точок.



- - - Початковий шлях (так само, як кінцевий шлях, якщо ніяких змін)  
 \_\_\_\_\_ Остаточний шлях після модифікації  
 ..... Шляхи, отримані в ході проміжних точок

Рисунок 2.14 – П'ять БЛА кожен з чотирма точками у навколишньому середовищі з перешкодами

### 2.3 Алгоритм організації доставки термінових вантажів за допомогою безпілотної авіації

На сьогодні, у сучасній науковій літературі виконано кілька досліджень в області аналізу і порівняння роботи алгоритмів найкоротших шляхів.

Наприклад, в статті [10] описуються дослідження з оцінки роботи алгоритму Дейкстри, евристичний і Генетичний алгоритми. У цій роботі використовували 6 реальних транспортних мереж для оцінки алгоритмів

найкоротшого шляху. Алгоритми були написані на мовах програмування Visual Basic, Visual C ++, MapObject. Програми розроблялися для знаходження групи найкоротших шляхів типу "від одного до одного", "від одного до всіх" і "від всіх до всіх". Алгоритми були реалізовані, використовуючи дані з різною кількістю вузлів і зв'язків.

В результаті тестування отримали, що:

- час виконання згаданих алгоритмів залежить від умов завдання і числа вузлів в реальних транспортних мережах;
- коли число вузлів і складність завдання збільшуються, Евристичний метод працює краще, ніж інші;
- для невеликого числа вузлів і складних умов завдання евристичного методу (Генетичний алгоритм) працює краще, ніж інші;
- для невеликого числа вузлів і легких умов завдання алгоритм Дейкстри застосовувати вигідніше, ніж інші.

В роботі [28] розглядалися такі алгоритми найкоротшого шляху, як алгоритм Йена, алгоритм Флойда і алгоритм Дейкстри. Ці методи були реалізовані в системі управління базами даних FoxPro 6.0. Однак, тут було прийнято апріорі, що алгоритм Дейкстри найкращий для шляхів між парою вершин мережі, Алгоритм Флойда кращий для вирішення завдань оптимізації шляхів між усіма парами вершин в мережі, алгоритм Йена найкращий для знаходження декількох найкоротших шляхів між двома вершинами транспортної мережі.

У дослідженнях, описаних в статті [30], алгоритми найкоротших шляхів були реалізовані в Borland Delphi 6. Але тут не виконувалося обґрунтування вибору найкращого алгоритму для знаходження найкоротшого (оптимального) шляху на транспортній мережі.

Якщо звернутися до глави "8.7 Завдання про найкоротший шлях і її варіанти" з книги І.В. Романовського [12], то тут розглядаються для вирішення завдання про найкоротшому шляху алгоритми Беллмана,

Уоршелла-Флойда, Дейкстри, Дейкстри-Грибова і Левіта. В ході розгляду цих методів були зроблені такі висновки:

«... Перевага алгоритму Уоршелла-Флойда полягає в тому, що він придатний для більшості графів з ребрами позитивної довжини, відмінний варіант для вирішення завдання в загальному випадку, без особливих умов. Недоліком алгоритму є необхідність кілька разів аналізувати одну і ту ж вершину, що призводить до зайвих ітерацій, а отже - до витрат часу і ресурсів ПК. Крім того, немає перевірки на можливість наявності в графі негативних ребер, що може привести до ускладнень в програмній реалізації...»

«... Ефективність методу Дейкстри істотно залежить від того, як організований пошук вершини з найменшою поточною відстанню ...»

«... Не дивлячись на дуже високу ефективність методу Дейкстри-Грибова, в практичних завданнях з «географічною» системою відстаней ще більш ефективний метод, належить Б.Ю.Левіту ... У порівнянні з методом Дейкстри метод Левіта програє на тому, що деякі вершини доводиться обробляти повторно, а виграє на більш простих алгоритмах включення і виключення вершин з безлічі  $M1$ . Експерименти показують, що для графів з «геометричним» походженням, тобто для графів, побудованих на основі транспортних мереж і реальних відстаней, метод Левіта виявляється найбільш швидким. Він виграє і за розміром програми. Метод Левіта має ще й ту перевагу перед методом Дейкстри, що він застосовується в разі негативних довжин дуг (адже «довжина дуги» - це просто назва, яке дає нам корисні асоціації з реальністю) ... »

Можна зробити висновок, що при проведенні досліджень не можна охопити відразу всі існуючі алгоритми розв'язання задачі. Найчастіше виконується порівняння декількох найбільш популярних і найбільш простих методів для вирішення тих чи інших видів завдань при заданих умовах.

У рамках даної роботи обрано:

- Алгоритм Дейкстри для вирішення транспортної задачі на мінімум часу;
- Алгоритм Беллмана-Форда для вирішення транспортної задачі на мінімум вартості;
- Метод потенціалів для вирішення транспортної задачі з обмеженнями на пропускну здатність.

Розглянемо їх детальніше.

У такій транспортній задачі вирішальну роль грає не вартість перевезень, а час, який витрачається на доставку вантажу. Оптимальним планом вважається план, який мінімізує час перевезень[12]. Є  $m$  пунктів відправлення, із запасами зерна  $a_i$ ,  $n$  пунктів призначення з потребами  $b_j$ .

$$\sum_{j=1}^n b_j = \sum_{i=1}^m a_i.$$

Ця задача закритого типу, тобто

Задана матриця  $T = [t_{ij}]_{m \times n}$ , де  $t_{ij}$  - час необхідний для перевезення вантажу з пункту  $i$  у пункт  $j$ .

Необхідно вибрати серед допустимих такий план  $X = [x_{ij}]_{m,n}$ , що

$$\left. \begin{array}{l} \sum_{i=1}^m x_{ij} = b_j, \forall j \in \overline{1, n} \\ \sum_{j=1}^n x_{ij} = a_i, \forall i \in \overline{1, m} \end{array} \right\}$$

і вантажі будуть доставлятися за цим планом за мінімальний час  $T_{\min}$ .

Кожному допустимому плану  $X = [x_{ij}]_{m,n}$  відповідає деякий набір  $\{t_{ij}\}_X$ , що складається з елементів матриці  $T = [t_{ij}]_{m \times n}$ , відповідних позитивним компонентам  $x_{ij}$  плану  $X$ . Тобто  $t_{ij}$  включається в набір, якщо здійснюється перевезення з пункту  $i$  в пункт  $j$ . [13]

Час  $t_X$ , необхідний для виконання плану  $X$ , визначається наступним чином  $t_X = \max\{t_{ij}\}_X$ .

Тоді час, необхідний для реалізації оптимального плану  $X^*$ .

$$t_{X^*} = \min_X t_X = \min_X (\max\{t_{ij}\}_X)$$

Алгоритм відшукування оптимального рішення

Даний алгоритм складається з двох етапів:

Попередній крок.

Будуємо допустимий план за методом північно-західного кута або мінімального елемента  $X_0$ .

Загальний крок.

Переглядаємо всі  $t_{ij}$ , відповідні позитивним  $x_{ij}$  і вибираємо з них

найбільше  $t_{ij}' = \max_{x_{ij}>0} \{t_{ij}\}$  і викреслюємо всі клітини, для яких  $t_{ij} \geq t_{ij}'$ ,  $\forall x_{ij} = 0$ .

Далі виробляємо виправлення плану  $X_0$ , для чого прагнемо звернути 0

в 0 перевезення  $x_{ij}'$ , відповідну  $t_{ij}' = \max_{x_{ij}>0} \{t_{ij}\}$  (у тій же клітині). Якщо це вдається, то, природно, зменшується час, необхідний на реалізацію нового допустимого плану  $X_1$ . Для побудови плану  $X_1$  будується цикл як і в методі потенціалів.

Першою клітиною негативного напівланцюга беремо клітку з  $t_{ij}'$ , іншими клітинами негативного напівланцюга вибираємо клітини з  $x_{ij} > 0$ , клітинами позитивного напівланцюга беремо клітини з  $t_{ij} < t_{ij}'$ .

Потім переміщаємо мінімальний елемент  $\theta$  негативного напівланцюга в позитивний. Якщо вдається  $x_{ij}'$  звернути в 0, то реалізація нового плану зажадає меншого часу[14].

Загальний крок продовжуємо повторювати до тих пір, поки стане неможливим звернення до 0 усього перевезення  $x_{ij}'$  з клітки з максимальним часом  $t_{ij}'$ .

Складаємо початкове опорне рішення  $X_1$  за методом північно-західного кута. Максимальна цільова функція  $T(X_1) = \max \{10, 8, 5, 12, 4\} = 12$  досягається в клітці (3,4). Перекреслюємо клітку (4,1), в якій час доставки вантажу  $t_{41} = 15$  більше, ніж  $T(X_1) = 12$ . [15]

Таблиця 2.1

$A_i \backslash B_j$	20	30	40	60
20	10 <b>20</b>	6	3	2
30	5	- 8 <b>30</b>	7	+ 4
50	2	+ 4	5 <b>40</b>	- 12 <b>10</b>
50	<del>15</del>	5	9	4 <b>50</b>

Для поліпшення рішення розвантаження звільнимо клітку (3,4) за допомогою циклу (3,4), (2,4), (2,2), (3,2). Означаючи цикл, знайдемо  $Q = \min \{10, 30\} = 10$ . Здійснивши зрушення по циклу, отримаємо друге опорне рішення:

$$T(X_2) = \max \{10, 8, 4, 5, 4\} = 10,$$



Досягається в клітці (1,1). Перекреслюємо клітку (3,4), тому що час  $t_{34} = 12$  більше, ніж  $T(X_2) = 10$ [16]

Таблиця 2.2

$B_j \backslash A_i$	20	30	40	60
20	- 10 <b>20</b>	+ 6	3	2
30	+ 5	- 8 <b>20</b>	7	4 10
50	2	4 <b>10</b>	5 <b>40</b>	<del>12</del>
50	<del>15</del>	5	9	4 <b>50</b>

Розвантажуюємо клітку (1,1) за допомогою циклу (1,1), (1,2), (2,2), (2,1).

Означаючи цикл, знайдемо:

$$Q = \min \{20, 20\} = 20.$$

Здійснивши зрушення по циклу, отримаємо третє опорне рішення  $X_3$ .

Максимум цільової функції на цьому опорному рішенні:

$$T(X_3) = \max \{6, 5, 4, 4, 5, 4\} = 6 \text{ і досягається в клітці (1,2).}$$

Перекреслюємо клітини (1,1) (2,2) (2,3) і (4,3) у них час більше, ніж  $T(X_3) =$

6

Таблиця 2.3

$B_j \backslash A_i$	20	30	40	60
20	10	- 6 <b>20</b>	+ 3	2
30	5 <b>20</b>	8	7	4 <b>10</b>
50	2	+ 4 <b>10</b>	- 5 <b>40</b>	12
50	15	5	9	4 <b>50</b>

Розвантажимо клітку (1,2) за допомогою циклу (1,2), (1,3), (3,3), (3,2).

Означаючи цикл, знайдемо:

$$Q = \min \{20, 20\} = 20$$

Таблиця 2.4

$B_j \backslash A_i$	20	30	40	60
20	10	6	3 <b>20</b>	2
30	5 <b>20</b>	8	7	4 <b>10</b>
50	2	4 <b>30</b>	5 <b>20</b>	12
50	15	5	9	4 <b>50</b>

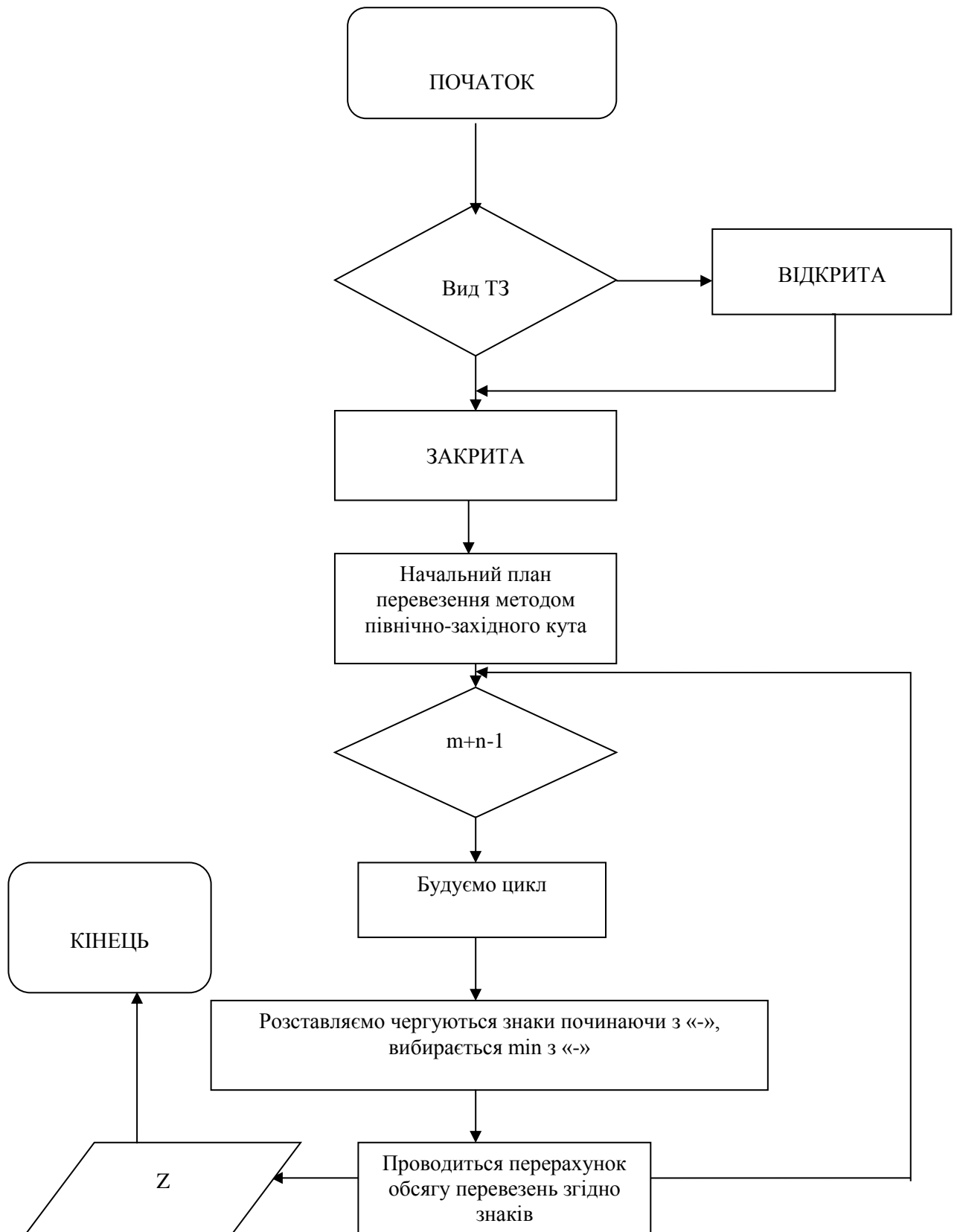


Рисунок 2.15 – Алгоритм рішення транспортної задачі на мінімум часу[17]

Здійснюємо зрушення по циклу, отримаємо 4 опорне рішення  $T(X_4)$ .

$T(X_4) = \max \{5, 4, 4, 5, 4\} = 5$  і досягається в клітинах (2,1) і (3,3).

Перекреслюємо клітини (1,2) і (4,2) у яких час перевезень не менш  $t_{21} = 5$ . За допомогою, клітин що залишилися не викресленими розвантажити клітини (2,1) і (3,3) не вдається, тому  $X_4$  є оптимальним рішенням.

$$\text{Відповідь: } \min T(X)=5 \text{ при } X^* = \begin{pmatrix} 0 & 0 & 20 & 0 \\ 20 & 0 & 0 & 10 \\ 0 & 30 & 20 & 0 \\ 0 & 0 & 0 & 50 \end{pmatrix}$$

Алгоритм рішення наведено на рис. 2.14

Транспортні задачі без обмежень на практиці зустрічаються рідко. Набагато частіше, при вирішенні транспортних задач, доводиться мати справу з обмеженнями пропускної здатності в тих чи інших клітинах (в задачах, представлених в матричній формі) або ланок (в задачах, представлених в мережевій формі)[18].

Пропускною здатністю клітин або ланок називається максимально можлива кількість перевезень, яку можна перевести в даній клітині або по ланці.

Рішення транспортних задач з обмеженнями проводиться аналогічно рішення завдання без обмежень. Однак є ряд особливостей.

Для транспортної задачі, представленої в матричній формі з обмеженнями пропускної здатності, необхідно знайти оптимальний план, при якому буде виконуватися умова найменшого сумарного обсягу тонно-кілометрової роботи. Для цього необхідно виконати наступні дії:

- 1) скласти математичну модель задачі;
- 2) скласти початковий план, перевірити за умовою "виродження", розрахувати сумарний обсяг тонно-кілометрової роботи;

3) вирішити задачу методом потенціалів, розрахувати сумарний обсяг тонно-кілометрової роботи оптимального плану;

4) порівняти початковий і оптимальний варіанти.

Необхідно побудувати оптимальний план перевезень зерна з п'яти портів  $A_i$  ( $i = 1,2,3,4,5$ ), до дев'яти великих споживачів, що мають під'їзні шляхи  $B_j$  ( $j = 1,2, \dots, 9$ ) [19].

Визначити обсяг тонно-кілометрової роботи початкового та оптимального планів перевезення вантажів.

Вихідні дані:

Дані про наявність ресурсів на п'яти складах  $A_i$  наведені в таблиці 2.5, дані про розміри прибуття вантажу  $B_j$  на дев'ять магазинів – у таблиці 2.6.

Таблиця 2.5 – Ресурси складів  $A_i$  (рядки матриці)

Номер складу	Значення
$A_1$	150
$A_2$	160
$A_3$	400
$A_4$	150
$A_5$	140
Всього:	1000

Таблиця 2.6 – Обсяг потреби  $B_j$  одержувача (стовпці матриці)

Номер магазину	Значення
$B_1$	135
$B_2$	105
$B_3$	95
$B_4$	115
$B_5$	85
$B_6$	105
$B_7$	90
$B_8$	135

$B_9$	135
Всього:	1000

Рішення:

Відстань перевезення від кожного  $i$ -го складу відправлення до кожного  $j$ -го магазину призначення зазначено у правому верхньому куті кожної клітини матриці. У лівому верхньому куті ряду клітин матриці вказані обмеження пропускної здатності.

Умовою завдання встановлено, що розмір всіх ресурсів у відправників дорівнює загальній потребі одержувачів [20]:

$$\sum_i A_i = \sum_j B_j$$

З урахуванням отриманих умов необхідно знайти такі невід'ємні значення величин обсягів перевезень  $x_{ij}$ , при яких сума добутків значень критерію  $C_{ij}$  на розмір перевезень буде мінімальною, тобто [21]

$$\sum_{i=1}^n \sum_{j=1}^n C_{ij} \cdot x_{ij} \rightarrow \min$$

Спочатку будується початковий план базисного варіанту способом найменшого значення критерію.

Будь-який допустимий план є оптимальним тоді і тільки тоді, коли кожному рядку і кожній колонці матриці можуть бути присвоєні деякі числа  $U_i$  і  $V_j$ , звані потенціалами і, які відповідають умовам [22]:

$$V_j - U_i \leq C_{ij} \text{ для } x_{ij} = 0; \quad (2.53)$$

$$V_j - U_i = C_{ij} \text{ для } d_{ij} > x_{ij} > 0, \quad (2.54)$$

$$V_j - U_i \geq C_{ij} \text{ для } x_{ij} = d_{ij}, \quad (2.55)$$

де  $V_j$  - потенціал  $j$ -го стовпчика;

$U_i$  - потенціал  $i$ -го рядка;

$C_{ij}$  - відстань перевезення від  $i$ -го постачальника до  $j$ -го споживача;

$x_{ij}$  - кореспонденція (розміри перевезень) від  $i$ -го постачальника до  $j$ -го споживача;

$d_{ij}$  - величина пропускної здатності  $ij$  клітини.

Присвоєння потенціалів починають з рядка, в якому серед базисних клітин є максимальна відстань. Цьому рядку можна присвоїти будь-який позитивний потенціал, наприклад, 100. Потім, використовуючи умову оптимальності (4.2), знаходять потенціали інших рядків і стовпців за формулами [23]:

для  $j$ -го стовпця

$$V_j = U_i + C_{ij};$$

для  $i$ -го рядка

$$U_i = V_j - C_{ij}.$$

Кореспонденція поліпшення плану знаходиться з наступного виразу:

$$x_{пол} = \min[x_{ijнар}(d_{ij} - x_{ij})_{ненар}]$$

$A_i$	$B_j$									
	B1=1 35	B2=1 05	B3= 95	B4=1 15	B5= 85	B6=1 05	B7= 90	B8=1 35	B9=1 35	$U_i$
A1=1 50	- 90 45 x	30	100	110  1+40	150  x	30 50 30	+ 60 75 x	80	90	100
A2=1 60	+ 10 80 x	40	45	50  1+20	- 25 80 x	70  1+10	30 15	30	10 30	180
A3=4 00	10 20 10	35 105 x	80  1+2 0	160	90  1+2 5	+ 80 • 1+90	- 70 15 x	40 135 x	60 135 x	90

A4=1 50	50	5	40	30 95	120	40 55	75	30	40 20	220
A5=1 40	15	15 25	10 95	20 35 20	+ 25 5	- 80 20	20	70	90	180
V <sub>j</sub>	190	125	190	250	205	260	160	130	150	

$F(x) = 45 \cdot 90 + 30 \cdot 50 + 75 \cdot 60 + 80 \cdot 10 + 80 \cdot 25 + 10 \cdot 20 + 105 \cdot 35 + 15 \cdot 70 + 135 \cdot 40 + 135 \cdot 60 + 95 \cdot 30 + 55 \cdot 40 + 95 \cdot 10 + 20 \cdot 35 + 5 \cdot 25 + 20 \cdot 80 = 39700$  грош.

од.

$80 - 70 + 60 - 90 + 10 - 25 + 25 - 80 = -90 < 0$  – цикл підходить

$r = \{15; 45; 80; 20\} = 15$

A <sub>i</sub>	B <sub>j</sub>									U <sub>i</sub>
	B1=1 35	B2 =10 5	B3= 95	B4=1 15	B5=8 5	B6 =1 05	B7= 90	B8=1 35	B9=1 35	
A1=1 50	- 90 30 x	+ 30 • 1+8 5	100	110  1+40	150	30 50 30	60 90 x	80	90  1+40 1+50	100
A2=1 60	+10 95 x	40	45	50  1+20	- 25 65 x	70  1+ 10	30 15	30  1+10	10 30	180
A3=4 00	10 20 10	- 35 105 x	80	160	90	+ 80 15 x	70	40 135 x	60 135 x	180
A4=1 50	50	5	40	30 95	120	40 55	75	30	40 20	220



				x		x				
A5=1 40	15	15 25	10 95 x	20 35 20 x	+ 25 20 x	- 80 5 x	20	70	90	180
V <sub>j</sub>	190	215	190	250	205	260	160	220	240	

$$F(x) = 39700 - 90 \cdot 15 = 38350 \text{ грош.од.}$$

$$30 - 90 + 10 - 25 + 25 - 80 + 80 - 35 = -85 < 0 - \text{цикл підходить}$$

$$r = \{30; 65; 5; 105\} = 5$$

A <sub>i</sub>	B <sub>j</sub>									U <sub>i</sub>
	B1=1 35	B2 =10 5	B3= 95	B4= 115	B5= 85	B6= 105	B7=9 0	B8=1 35	B9=1 35	
A1=1 50	- 90 25 x	+ 30 5 x	100	110	150	30 50 30	60 90 x	80	90	100
A2=1 60	+ 10 100 x	40	45	50	- 25 60 x	70	30 15	30	10 30	180
A3=4 00	10 20 10	- 35 100 x	80 1+1 5	160	+ 90 • 1+2 0	80 20 x	70	40 135 x	60 135 x	95
A4=1 50	50 1+5	5	40 1+1 5	30 95 x	120	40 55 x	75	30	40 20	135
A5=1 40	15	15 25	10 95 x	20 35 20	25 25 x	80	20	70	90	180

$V_j$	190	130	190	165	205	175	160	135	155	
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	--

$$F(x) = 38350 - 85 \cdot 5 = 37925 \text{ грош.од.}$$

$$90 - 25 + 10 - 90 + 30 - 35 = -20 < 0 - \text{цикл підходить}$$

$$r = \{60; 25; 100\} = 25$$

$A_i$	$B_j$									$U_i$
	$B1=$ 135	$B2=1$ 05	$B3=$ 95	$B4=1$ 15	$B5=$ 85	$B6=1$ 05	$B7=$ 90	$B8=1$ 35	$B9=1$ 35	
$A1=1$ 50	90	30 30 x	100	110	150	30 50 30	60 90 x	80	90	105
$A2=1$ 60	10 125 x	40	45	50	25 35 x	70	30 15	30	10 30	165
$A3=4$ 00	10 20 10	35 75 x	80	160	90 25 x	80 20 x	70	40 135 x	60 135 x	100
$A4=1$ 50	50	5	40	30 95 x	120	40 55 x	75	30	40 20	140
$A5=1$ 40	15	15 25	10 95 x	20 35 20	25 25 x	80	20	70	90	165
$V_j$	175	135	175	170	190	180	165	140	160	

План оптимальний.

$$F(x) = 37925 - 20 \cdot 25 = 37425 \text{ грош.од.}$$

$$\text{Відповідь: } F(x)_{\text{поч.}} = 39700 \text{ грош.од.}; F(x)_{\text{опт.}} = 37425 \text{ грош.од.}$$

Метод найменшої вартості. При цьому методі на кожному кроці побудови опорного плану першою заповнюється та клітина, яка залишилася, у таблиці, яка має найменший тариф. Якщо така клітина не єдина, то заповнюється кожна з них [25-28]. У діагональному методі не враховуються величини тарифів, у методі ж найменшої вартості ці величини враховуються, і часто останній метод призводить до плану з меншими загальними витратами, хоча це й не обов'язково.

Складемо умову. Припустимо, що з трьох портів  $A_1, A_2, A_3$  необхідно доставити вантаж у п'ять точок  $B_1, B_2, B_3, B_4, B_5$ . Потрібно закріпити точки доставки на маршруті так, щоб загальна сума витрат на перевезення була мінімальною.

Числові дані завдання представлені в наступній таблиці 2.7:

Таблиця 2.7 – Числові данні

Порт	Точки доставки вантажу					Обсяг вивозу, т.
	$B_1$	$B_2$	$B_3$	$B_4$	$B_5$	
	Вартість перевозки 1т. вантажу, грн.					
$A_1$	3	3	2	4	2	40
$A_2$	6	2	3	1	7	150
$A_3$	4	5	2	8	4	100
Обсяг вивозу, т.	20	80	90	60	40	290

Нехай  $x_{ij}$  - кількість одиниць вантажу, яке необхідно доставити від  $i$ -го складу до  $j$ -ої точки призначення.

Тоді сумарні витрати на перевезення  $Z$  складуть:

$$Z = 3 \cdot x_{11} + 3 \cdot x_{12} + 2 \cdot x_{13} + 4 \cdot x_{14} + 2 \cdot x_{15} + 6 \cdot x_{21} + 2 \cdot x_{22} + 3 \cdot x_{23} + 1 \cdot x_{24} + 7 \cdot x_{25} +$$

$$+ 4 \cdot x_{31} + 5 \cdot x_{32} + 2 \cdot x_{33} + 8 \cdot x_{34} + 4 \cdot x_{35} \rightarrow \min \begin{cases} x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 40 \\ x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 150 \\ x_{31} + x_{32} + x_{33} + x_{34} + x_{35} = 100 \\ x_{11} + x_{21} + x_{31} = 20 \\ x_{12} + x_{22} + x_{32} = 80 \\ x_{13} + x_{23} + x_{33} = 90 \\ x_{14} + x_{24} + x_{34} = 60 \\ x_{15} + x_{25} + x_{35} = 40 \\ x_{ij} \geq 0, i = \overline{1,3}; j = \overline{1,5} \end{cases}$$

### Алгоритм рішення

1. Визначаємо характер транспортної задачі. Модель транспортної задачі називають закритою, якщо  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ . В іншому випадку модель

транспортної задачі називаю відкритою. Якщо  $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$ , то відкрита

транспортна задача зводиться до закритої шляхом введення фіктивного

споживача з об'ємом потреб  $b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$  і вартостями перевезень рівними

0. Якщо  $\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$ , то відкрита транспортна задача зводиться до закритої

шляхом введення фіктивного споживача з об'ємом потреб  $a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$  і

вартостями перевезень рівними 0.

2. Складемо перший опорний план методом «мінімальної вартості».

3. Перевіримо отриманий опорний план на невиродженість [29]. План називається невиродженим, якщо виконується умова  $k = m + n - 1$ , де  $k$  - це число заповнених клітинок,  $m$  - число постачальників,  $n$  - число споживачів. Якщо опорний план вироджений (умова не виконується), необхідно перейти від нього до невиродженого опорного плану. Для цього незаповнену клітинку з мінімальною вартістю перевезень заповнюємо нулем, але без утворення замкнутого циклу з опорними вершинами в заповнених клітинках.

4. Визначимо потенціали постачальників  $u_i$  і споживачів  $v_j$  з рівнянь  $u_i + v_j = c_{ij}$ . При цьому передбачається, що  $u_1 \equiv 0$ .

5. Перевіримо опорний план на оптимальність, обчисливши оцінки вільних осередків:  $s_{ij} = c_{ij} - (u_i + v_j)$ . Якщо  $s_{ij} \geq 0$ , то знайдений план оптимальний. При цьому якщо якась оцінка,  $s_{ij} = 0$  то оптимальний план неєдиним. У разі якщо всі оцінки  $s_{ij} > 0$ , то оптимальний план єдиний.

6. Якщо яка-небудь з оцінок  $s_{ij} \leq 0$ , то план неоптимальний і необхідно зробити перерозподіл поставок (зробити завантаження вільного осередку з негативною оцінкою) [20].

7. Кроки алгоритму 4-6 повторюємо до тих пір, поки не буде отриманий оптимальний опорний план.

Задача є закритою, тому що  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ .

Перший опорний план:

	20	80	90	60	40
40	3	3	240	4	2
150	6	280	3	160	710
100	420	5	250	8	430

Так як  $k = n + m - 1$  (тобто опорний план невироджений). Визначимо потенціали портів відправки  $u_i$  і точок доставки  $v_j$ .

	20	80	90	60	40	
40	3	3	240	4	2	0
150	6	280	3	160	710	3
100	420	5	250	8	430	0
	4	-1	2	-2	4	

Оцінки вільних осередків:

$$S_{11}=-1;$$

$$S_{12}=4;$$

$$S_{14}=6;$$

$$S_{15}=-2;$$

$$S_{21}=-1$$

$$S_{23}=-2;$$

$$S_{32}=6$$

$$S_{34}=10$$

Отриманий опорний план неоптимальний, тому що серед оцінок вільних осередків є негативні, візьмемо саму мінімальну з них, це  $S_{15} = -2$ ; і  $S_{23} = -2$ ; ми використовуємо  $S_{15} = -2$ .

Зробимо перерозподіл поставок. Для цього побудуємо замкнутий цикл.

	20	80	90	60	40	
40	3	3	«-» 240	4	2 «+»	0
150	6	280	3	160	7 10 «-»	3
100	420	5	«-» 250	8	4 30	0
	4	-1	2	-2	4	

Зробимо завантаження вільного осередку з негативною оцінкою. Поставку, передану по циклу, визначаємо як  $\min \{40, 30\} = 30$ ; Визначимо потенціали портів відправників  $u_i$  і точок доставки  $v_j$ . У результаті перетворень отримаємо наступну таблицю [21].

	20	80	90	60	40	
40	3	3	«-» 10	4	2 30 «+»	0
150	6	280	«+» 3	160	7 «-» 10	5
100	420	5	280	8	40	2
	2	-3	0	-4	2	

$$S_{11}=1;$$

$$S_{12}=6;$$

$$S_{14}=8;$$

$$S_{21}=-1$$

$$S_{23}=-2;$$

$$S_{32}=6;$$

$$S_{34}=10;$$

Отриманий опорний план неоптимальний, тому що серед оцінок вільних комірок є негативні, візьмемо саму мінімальну з них, це  $S_{23} = -2$ ; Зробимо перерозподіл поставок. Для цього побудуємо замкнутий цикл.

	20	80	90	60	40	
40	3 «+»	3	20 «-»	4	240	0
150	6	280	310 «+»	160	7	1
100	420	5	280	8	40	0
	4	1	2	0	2	

$$S_{11}=-1;$$

$$S_{12}=2;$$

$$S_{14}=4;$$

$$S_{21}=1;$$

$$S_{25}=4;$$

$$S_{32}=4;$$

$$S_{34}=8;$$

Отриманий опорний план неоптимальний, тому що серед оцінок вільних комірок є негативні, візьмемо саму мінімальну з них, це  $S_{11} = -1$ ;

Зробимо перерозподіл поставок. Для цього побудуємо замкнутий цикл.

	20	80	90	60	40	
40	30	3	2	4	240	0
150	6	280	3 10	160	7	2
100	420	5	280	8	4	1
	3	0	1	-1	2	

$$S_{12}=3;$$

$$S_{13}=1;$$

$$S_{14}=5;$$

$$S_{21}=1;$$

$$S_{25}=3;$$

$$S_{32}=4;$$

$$S_{34}=8;$$

Всі отримані оцінки невід'ємні, значить знайдений опорний план оптимальний і єдиний.

Оптимальний план розподілу:

$$X = \begin{pmatrix} 0 & 0 & 0 & 0 & 40 \\ 0 & 80 & 10 & 60 & 0 \\ 20 & 0 & 80 & 0 & 0 \end{pmatrix}$$

Для нього значення цільової функції (мінімальні транспортні витрати)

$$Z = 2 \cdot 40 + 2 \cdot 80 + 3 \cdot 10 + 1 \cdot 60 + 4 \cdot 20 + 2 \cdot 80 = 570 \text{ грн.}$$



## **Висновки до розділу**

У межах розділу визначено функціональні складові системи організації доставки термінових вантажів за допомогою безпілотної авіації, детально описано математичну складову побудови маршрутів БПЛА. Наведено алгоритми організації доставки термінових вантажів за допомогою безпілотної авіації з обґрунтуванням вибору.

## *ПРОЕКТНА ЧАСТИНА*

Кафедра ОАП				НАУ 20.02.20. 300 ПЗ			
Виконав	Петрук В.О.			Проектна частина	Літера	Аркуш	Аркушів
Керівник	Юн Г.М.				Д	90	18
Консульт.	Юн Г.М.				ФТМЛ 275 ОП-201М		
Н.Контр.	Жукова С.О.						
Зав. каф.	Юн Г.М.						

### 3.1 Структура організації доставки термінових вантажів за допомогою безпілотної авіації

Проект розроблений з використанням мови програмування C++ для використання середовища під керуванням операційної системи Windows.

C++ - компільована статично типізована мова програмування загального призначення.

Підтримує такі парадигми програмування як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування, забезпечує модульність, роздільну компіляцію, обробку винятків, абстракцію даних, оголошення типів (класів) об'єктів, віртуальні функції. Стандартна бібліотека включає, в тому числі, загальновживані контейнери і алгоритми. C++ поєднує властивості як високорівневих, так і низькорівневих мов. У порівнянні з її попередником - мовою C, - найбільшу увагу приділено підтримці об'єктно-орієнтованого і узагальненого програмування [31].

C++ широко використовується для розробки програмного забезпечення, будучи однією з найбільш популярних мов програмування. Область її застосування включає створення операційних систем, різноманітних прикладних програм, драйверів пристроїв, додатків для вбудованих систем, високопродуктивних серверів, а також розважальних додатків (ігор). Існує безліч реалізацій мови C++, як безкоштовних, так і комерційних і для різних платформ. Наприклад, на платформі x86 це GCC, Visual C++, Intel C++ Compiler, Embarcadero (Borland) C++ Builder та інші. C++ зробила величезний вплив на інші мови програмування, в першу чергу на Java і C#.

Синтаксис C++ успадкований від мови C. Одним з принципів розробки було збереження сумісності з C. Проте, C++ не є в строгому сенсі над безліччю C; безліч програм, які можуть однаково успішно транслюватися

як компіляторами C, так і компіляторами C++, досить велике, але не включає всі можливі програми на C.

Windows – узагальнююча назва операційних систем для ЕОМ, розроблених корпорацією Microsoft. Перші версії були не повноцінними операційними системами, а лише оболонками до ОС MS-DOS. На 2014 рік, за даними сайтів NetApplications та GoStats, Microsoft Windows встановлена більш як на 90% персональних комп'ютерів світу

Пакет Microsoft Windows включає стандартні застосунки такі як браузер (Internet Explorer), поштовий клієнт (Outlook Express), програвач (Windows Media Player). За допомогою технологій COM і OLE їхні компоненти можуть вбудовуватися в інші застосунки, у тому числі й сторонніх виробників. Ці продукти позиціонуються як безплатні і можуть бути вільно викачані з офіційного сайту Microsoft, проте для установки деяких з них необхідно мати робочу ліцензію Windows. Запуск цих програм під іншими операційними системами можливий за допомогою емуляторів середовища Windows. Існують також версії деяких з них, спеціально розроблені для інших операційних систем. Ці версії поступаються оригінальним версіям набором функцій і можливостей, а також частотою оновлення.

Навколо факту включення таких стандартних продуктів в ОС Windows розгортається багато суперечок і дискусій, оскільки це створює серйозні перепони для розповсюдження конкуруючих продуктів. Так, часто оспорується лідерство і ставлять під сумнів якість браузера Internet Explorer, пояснюючи його популярність входженням в пакет Windows і поганою обізнаністю користувачів про наявність альтернатив. А в березні 2004 року Європейська Комісія оштрафувала Майкрософт і зобов'язала створити для продажу в Європі версію Windows без Windows Media Player. Сама компанія Майкрософт продовжує відстоювати своє право включати свої продукти в пакет Windows і розробляє нові інтегровані продукти.

В даний час Windows встановлена на більшості персональних комп'ютерів і робочих станцій. Багато користувачів зробили вибір на користь Microsoft Windows, але дуже велика кількість зовсім не знайомі з альтернативами, такими як Mac OS Xx86, GNU/Linux, BSD і іншими системами. В Україні до початку 2000-х років майже всі персональні комп'ютери продавалися з перед встановленою системою Windows. Боротьба з розповсюдженням нелегальних копій програмних продуктів привела до появи інтересу до вільних операційних систем. Так, наприклад, стало можливим придбати персональний комп'ютер з перед встановленою безкоштовною операційною системою GNU/Linux, FreeBSD і іншими.

Технічне забезпечення – комплекс технічних засобів, призначених для роботи інформаційної системи, а також відповідна документація на ці засоби і технологічні процеси.

Комплекс технічних засобів становить:

- Комп'ютери будь-яких моделей;
- Пристрої збору, накопичення, обробки, передачі та виведення інформації;
- Пристрої передачі даних і ліній зв'язку;
- Оргтехніка й пристрої автоматичного знімання інформації;
- Експлуатаційні матеріали та ін.

Документацією оформляються попередній вибір технічних засобів, організація їх експлуатації, технологічний процес обробки даних, технологічне оснащення. Документацію можна умовно розділити на три групи:

- Загальносистемну, що включає державні та галузеві стандарти по технічному забезпеченню;
- Спеціалізовану, яка містить комплекс методик по всіх етапах розробки технічного забезпечення;

- Нормативно-довідкову, використовувану при виконанні розрахунків по технічному забезпеченню [9].

До теперішнього часу склалися дві основні форми організації технічного забезпечення (форми використання технічних засобів): централізована і частково або повністю децентралізоване.

Централізоване технічне забезпечення базується на використанні в інформаційній системі великих ЕОМ і обчислювальних центрів.

Децентралізація технічних засобів припускає реалізацію функціональних підсистем на персональних комп'ютерах безпосередньо на робочих місцях.

Перспективним підходом варто вважати, мабуть, частково децентралізований підхід – організацію технічного забезпечення на базі розподілених мереж, що складаються з персональних комп'ютерів і великій ЕОМ для зберігання баз даних, загальних для будь-яких функціональних підсистем.

Еволюція технічного забезпечення, яке включає в себе апаратні засоби, засоби комунікації, програмне забезпечення, проходить нерівномірно, стрибкоподібно. Розвиток комп'ютерної техніки поки відбувається в геометричній прогресії. Кожні чотири роки відбувається подвоєння продуктивності комп'ютерів.

Класифікація комп'ютерних технологій за типом користувацького інтерфейсу (як взаємодіє користувач технології з комп'ютером) - пакетні, діалогові, мережеві. У першому випадку користувач отримує тільки результати роботи технології, в інших він взаємодіє з нею на індивідуальному комп'ютері чи комп'ютері, підключеному до мережі ЕОМ.

Сучасні технічні засоби забезпечення управління інформаційними ресурсами за своїм складом і функціональним можливостям вельми різноманітні. Засоби обчислювальної техніки, засоби комунікаційної техніки, засоби організаційної техніки [6].

Комп'ютерна техніка призначена, в основному, для реалізації комплексних технологій обробки і зберігання інформації і є базою інтеграцією всіх сучасних технічних засобів забезпечення управління інформаційними ресурсами.

Комунікаційна техніка призначена, в основному, для реалізації технологій передачі інформації і припускає як автономне функціонування, так і функціонування в комплексі із засобами комп'ютерної техніки.

Організаційна техніка призначена для реалізації технологій зберігання, подання та використання інформації, а також для виконання різних допоміжних операцій в рамках тих чи інших технологій інформаційної підтримки управлінської діяльності.

Програмні засоби автоматизації офісу діляться на системні та прикладні програми.

До системних програм відносяться: тестові та діагностичні програми, антивірусні програми, операційні системи, командно-файлові процесори.

Тестові і діагностичні програми призначені для перевірки працездатності окремих вузлів комп'ютера і компонентів програмно-файлових систем і, можливо, для усунення виявлених несправностей.

Антивірусні програми призначені для виявлення і, можливо, усунення вірусних програм, що порушують нормальну роботу обчислювальної системи.

Операційні системи є основними системними програмними комплексами, які виконують такі основні функції: тестування працездатності обчислювальної системи і її настройку при первісному включенні; забезпечення синхронної та ефективної взаємодії всіх апаратних і програмних компонентів обчислювальної системи в процесі її функціонування; забезпечення ефективної взаємодії користувача з обчислювальною системою.

Операційні системи класифікуються наступним чином: одно користувачькі одно задачні системи (MS-DOS, DR-DOS); одно користувачькі

багато задачні системи (Windows 2007, 2010); багатокористувацькі системи (системи сімейства UNIX).

Командно-файлові процесори (оболонки) призначені для організації системи взаємодії користувача з обчислювальною системою на принципах, відмінних від реалізованих операційною системою, з метою полегшення роботи або надання додаткових можливостей (наприклад, Norton Commander) [7].

Прикладні програмні засоби забезпечення управлінської діяльності класифікуються наступним чином системи підготовки текстових документів; системи обробки фінансово-економічної інформації; системи управління базами даних; особисті інформаційні системи; системи підготовки презентацій; системи управління проектами; експертні системи та системи підтримки прийняття рішень; системи інтелектуального проектування і вдосконалення систем управління; інші системи.

Для функціонування розроблюваного програмного додатку обрана наступна конфігурація:

- процесор – Pentium IV 3.2 GHz ;
- пам'ять – від 1 Gb ;
- жорсткий диск від 80 Gb;
- відеокарта від 64 Mb, CD- ROM – від 48x;
- монітор – 19 " Samsung SyncMaster,
- принтер HP LaserJet 1100;
- клавіатура і миша Genius.

Дана конфігурація покликана забезпечити комфортну роботу з офісними додатками.

Наведена вище конфігурація – нижча від середнього рівня оснащення комп'ютерів користувачів, на даний момент часу, що говорить про доступність програми широкому колу користувачів.



Структуру системи організації доставки термінових вантажів за допомогою безпілотної авіації складають низка класів, до їх числа варто віднести:

клас-об'єкт "матриця, суміжності"(ADJACENT MATRIX) для G-графа

клас-об'єкт "динамічний масив"( DYNAMIC ARRAY)

клас-об'єкт "двостороння черга"

клас-об'єкт "СТЕК" (STACK) LIFO - "останнім зайшов, першим вийшов"

клас-об'єкт "черга" ((QUEUE) FIFO - "останнім зайшов, першим вийшов"

клас-об'єкт фіксований масив (FIXED BLOCK ARRAY)

клас-об'єкт пріоритетна черга (PRIORITY QUEUE) на базі бінарної купи

клас-об'єкт (КОНТРОЛ GRAPH EDIT) для створення графів (неорієнтованих або орграфов) в графічному уявленні.

Оскільки АРМ, як елемент автоматизованої логістичної системи, як правило, зорієнтований (принаймні на сучасному етапі) на користувачів, які мають недостатню підготовку для роботи з ПЕОМ, а робота в середовищі АРМ передбачає оперативну взаємодію в діалоговому режимі користувача з іншими елементами системи, важливої ролі набуває інтерфейс користувача в середовищі АРМ. Інтерфейс (interface) - це засіб взаємодії користувача із системою.

Цим терміном в інформатиці позначають досить широке коло понять:

- фізичний (апаратний);
- інтерфейс на рівні електронних компонентів;
- інтерфейс програміста (комплекс правил і погоджень про стикування програмних модулів);
- інтерфейс користувача (набір засобів діалогу, взаємодії програми з людиною).

Інтерфейс розробленої автоматизованої системи організації доставки термінових вантажів за допомогою безпілотної авіації виглядає наступним чином (рис. 3.1):

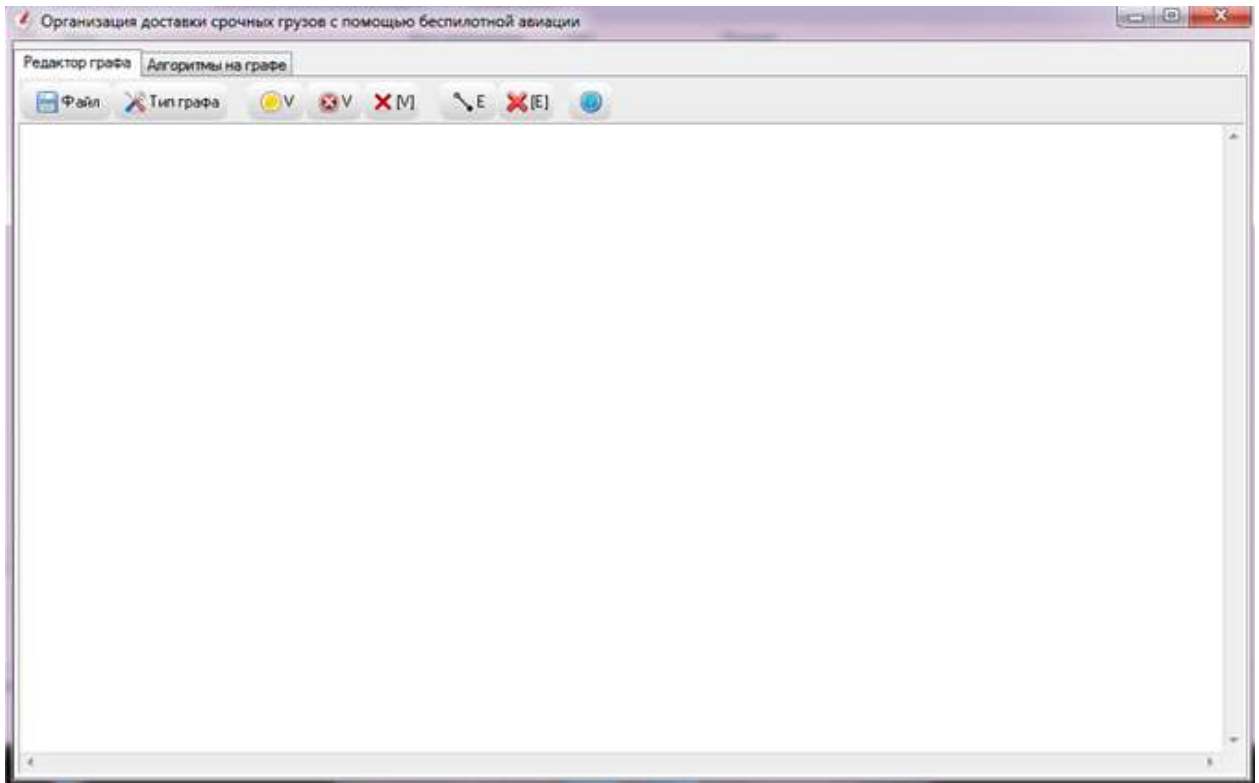


Рисунок 3.1 – Інтерфейс робочої автоматизованої системи організації доставки термінових вантажів за допомогою безпілотної авіації

У межах даної розробки вирішується транспортна задача про оптимальний план перевезень термінових вантажів зі складу до міста призначення. Потрібно досягти мінімальних витрат на перевезення. Шлях відображається на карті.

У центрі вікна програми розміщено карту доріг, яка представляється у вигляді графа. Вершинами є пункти призначення, а ребра графа – це шлях. На ребрах (шляхах) приписані їх характеристики: відстань (до наступного пункту), а також пропускна здатність в день.

Підприємство займається постачанням вакцин, медичних препаратів, біологічних речовин (крові).

У верхній частині екрану розміщено посилання де можна побудувати граф перевезення



За допомогою інтерфейсу людина керує роботою системи: видає завдання; відповідає на її запити й отримує інформацію про хід роботи програми. У деяких випадках інтерфейс застосовується і для оформлення результатів роботи. Основні елементи інтерфейсу - меню і діалогове вікно

Меню розробленої системи є простим у користуванні.

### **3.2 Основні модулі системи організації доставки термінових вантажів за допомогою безпілотної авіації**

Розглянемо основні модулі системи організації доставки термінових вантажів за допомогою безпілотної авіації.

Матриця суміжності задається процедурою:

```
class xAdjMatrix {
private:
    int**  mat;
    UINT  cnt;
    UINT  size;
    HANDLE heap;
public:

    xAdjMatrix(void): mat(NULL), cnt(0u), size(0u), heap(NULL) {}
    ~xAdjMatrix() {
        this->Clear();
    }

    xAdjMatrix(const xAdjMatrix&);
    xAdjMatrix& operator = (const xAdjMatrix&);
public:

    bool SetAlloc(HANDLE _heap, UINT num_size) {
        UINT icnt;
        heap = _heap;

        this->Clear();
    }
};
```

```

        mat      =      (int**)HeapAlloc(_heap,      SDATA_HEAP_NOLOCK,
(DWORD)(num_size * sizeof(int*)));
        if(mat == NULL)
            return false;

        for(icnt = 0u; icnt < num_size; ++icnt) {
            mat[icnt] = (int*)HeapAlloc(_heap, SDATA_HEAP_NOLOCK,
(DWORD)(num_size * sizeof(int)));
            if(mat[icnt] == NULL)
                break;
        }

        if(icnt != num_size) {
            for(UINT i = 0u; i < icnt; ++i)
                HeapFree(_heap, SDATA_HEAP_NOLOCK,
(LPVOID)mat[i]);

            HeapFree(_heap, SDATA_HEAP_NOLOCK, (LPVOID)mat);
            mat = NULL;
            return false;
        }

        size = num_size;
        this->Reset();
        return true;
    }

    HANDLE GetHeap(void) const {
        return heap;
    }

    void SetCount(UINT num) {
        cnt = num;
    }

```

### Сортування шела визначається як

```

void Sort(void) {
    int num = (int)cnt;
    for(int d = num >> 1; d > 0; d >>= 1) {
        for(int i = d; i < num; ++i) {
            for(int j = i; j >= d; j -= d) {
                if(arr[j - d] > arr[j]) {
                    T t      = arr[j - d];
                    arr[j - d] = arr[j];
                    arr[j]     = t;
                }
            }
        }
    }
}

```

### Виведення графу та його формування представлено, як

```

class xGraphApp {
private:
    xGraphPages  c_tabs;
    xGraphEdit  c_graph_edit;
    xToolBar    c_etoolbar;
    int         c_epos;
    xGraphView  c_graph_view;
    xToolBar    c_vtoolbar;
    xStatusBar  c_vstatus;
    xToolTip    c_tooltip;
public:

```

```

    xGraphApp(void);
    ~xGraphApp();
    xGraphApp(const xGraphApp&);
    xGraphApp& operator = (const xGraphApp&);
public:
    BOOL Create(HWND parent, int x, int y, int width, int height);
    void Destroy(void);

    void ReSize(int width, int height);
private:
    xImageButton* create_button(HWND parent, HWND hgraph, UINT id, int
y, int width, int height, const TCHAR* hlp = NULL);
    void set_ibutton(xImageButton* pbut, HINSTANCE inst, UINT
id, const TCHAR* str, xToolBar* bar);

```

Проведемо тестування системи організації доставки термінових вантажів за допомогою безпілотної авіації.

### **3.3 Тестування системи організації доставки термінових вантажів за допомогою безпілотної авіації**

Проведемо тестування розробленої системи організації доставки термінових вантажів за допомогою безпілотної авіації. Тестування здійснюємо на максимальному навантаженні, результати представлено на рис. 3.2 – 3.10.

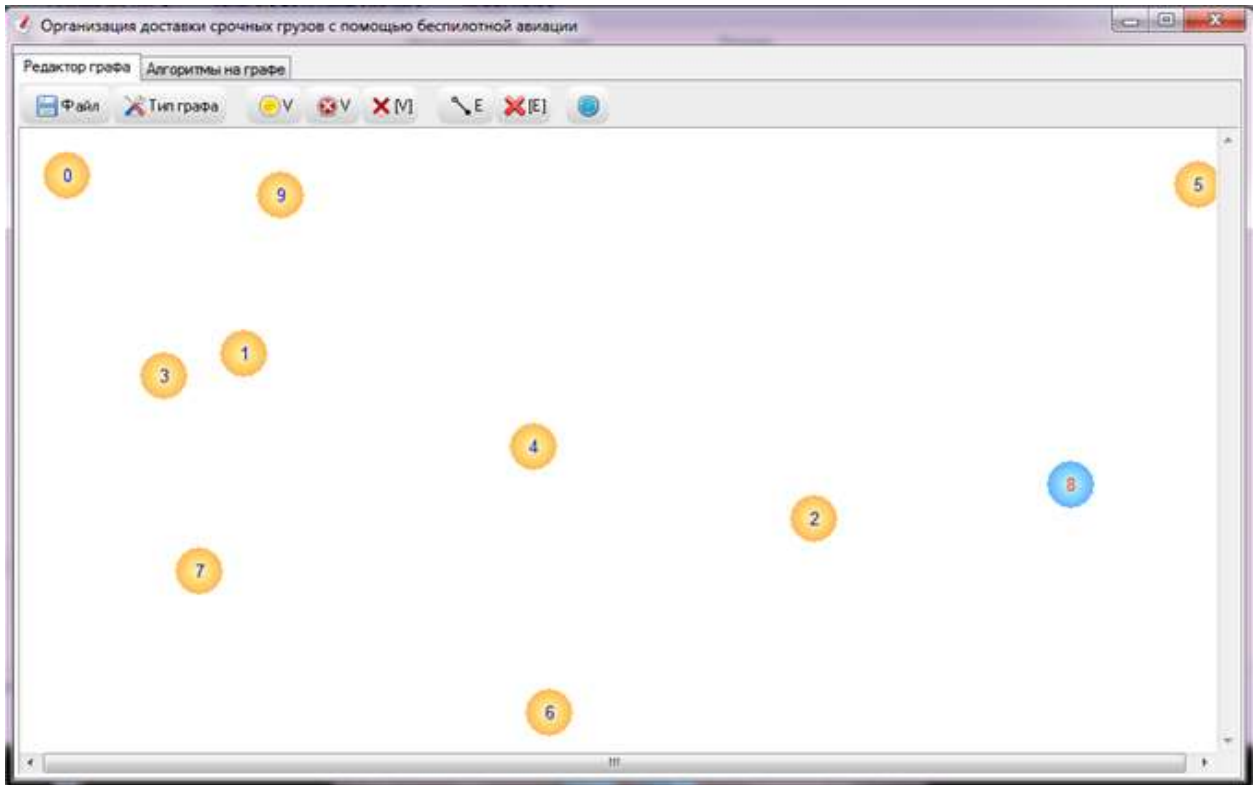


Рисунок 3.2 – Результат тестування. Побудова графу

На рис. 3.2 наведено результат винесення пунктів (вершин).

Далі встановлюємо відстані між пунктами (рис. 3.3)

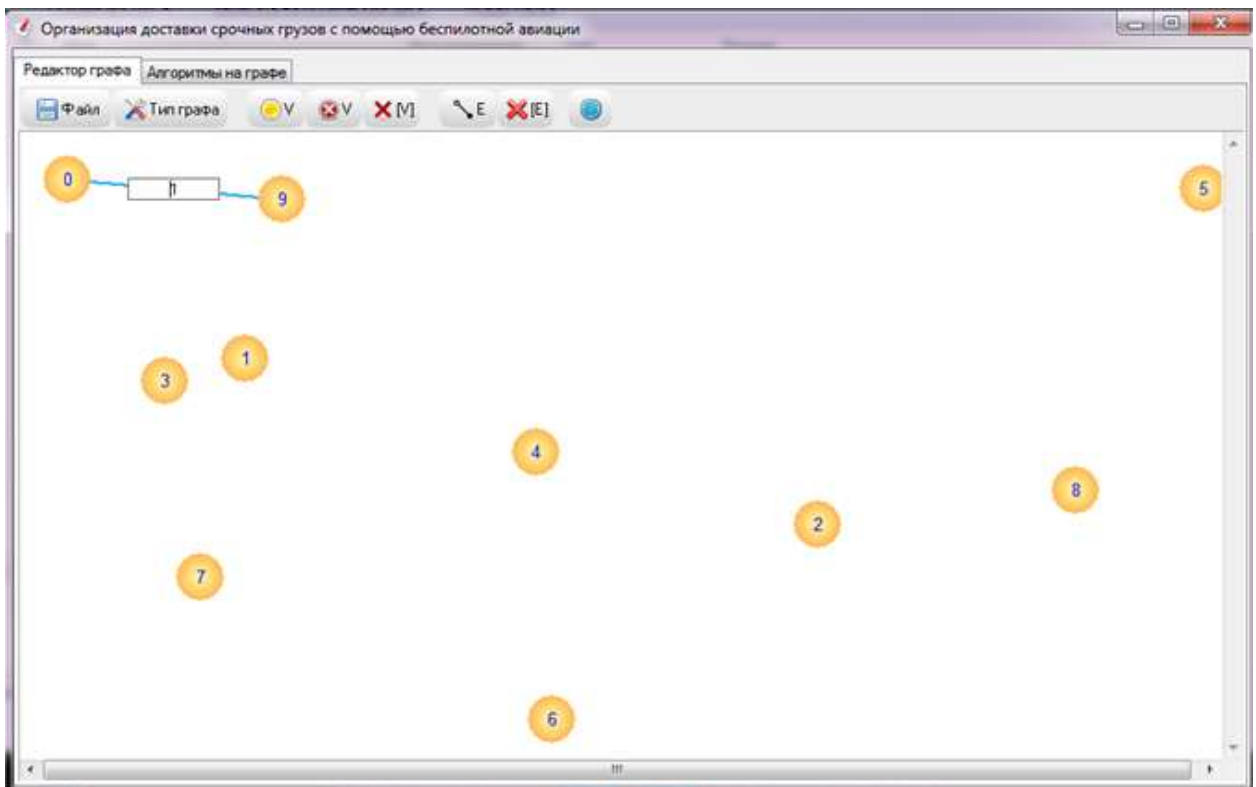


Рисунок 3.3 – Результат тестування. Встановлюємо відстань між пунктами

Здійснюємо формування маршруту за всіма пунктами, встановлюємо відстані (рис. 3.4).

Далі здійснюємо перехід на вкладку алгоритмів, та з випадаючого списку обираємо алгоритм побудови шляху перевезення термінових вантажів (рис 3.5).

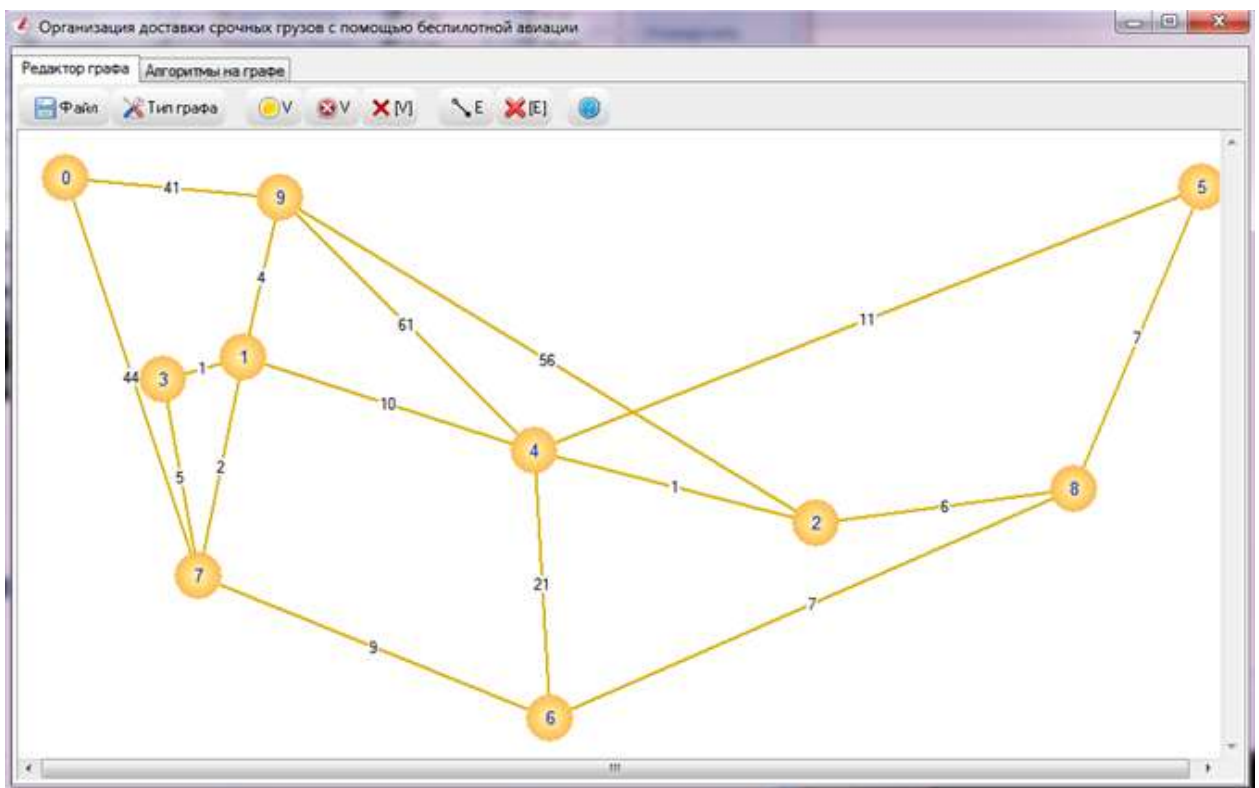


Рисунок 3.4 – Результат тестування. Відновлений граф

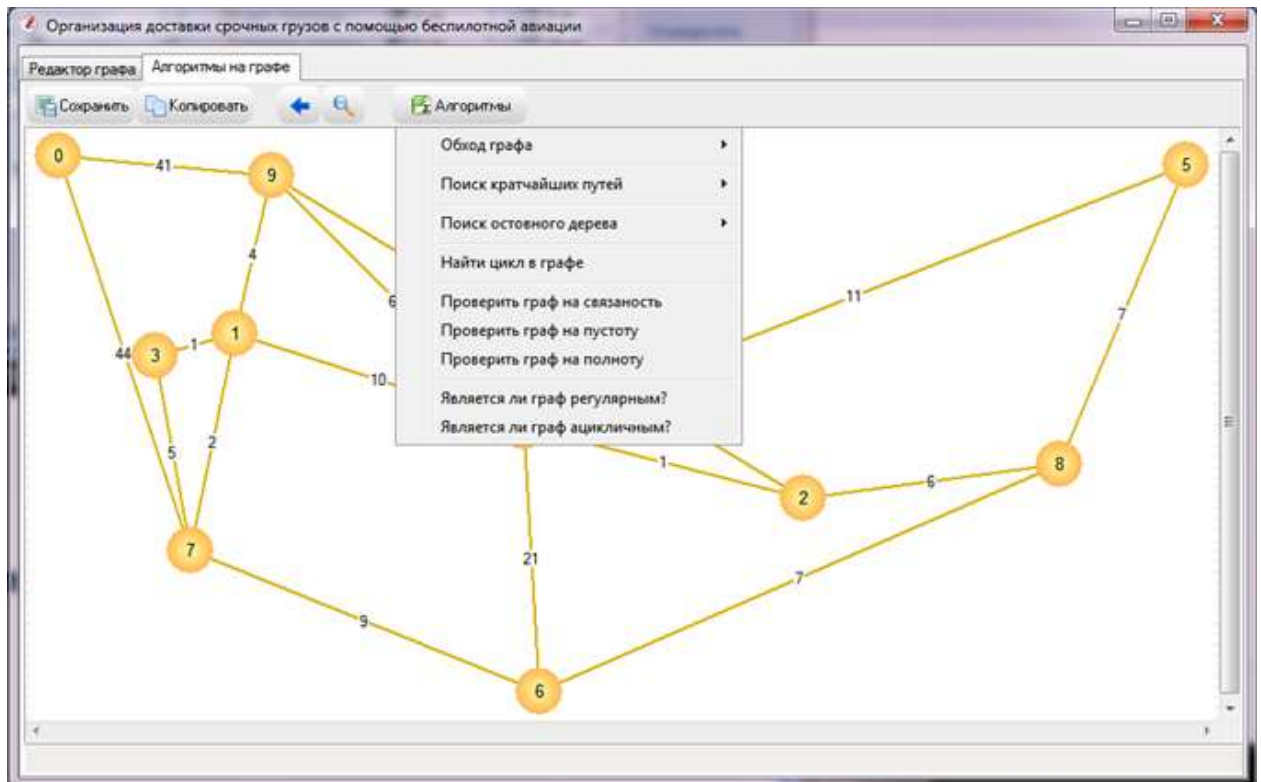


Рисунок 3.5 – Результат тестування. Відновлений граф

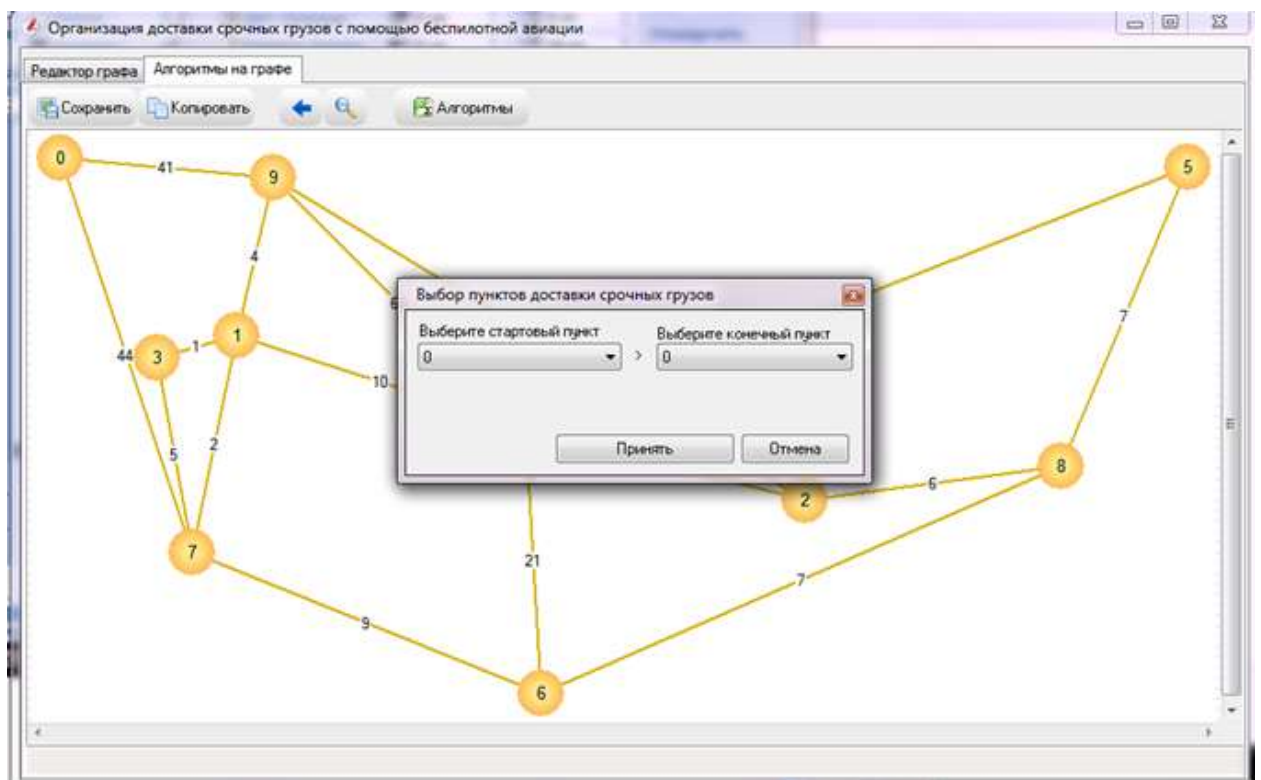


Рисунок 3.6 – Результат тестування. Відновлений граф

Здійснюємо вибір стартового пункту та обираємо номер пункту прибуття (рис. 3.7).



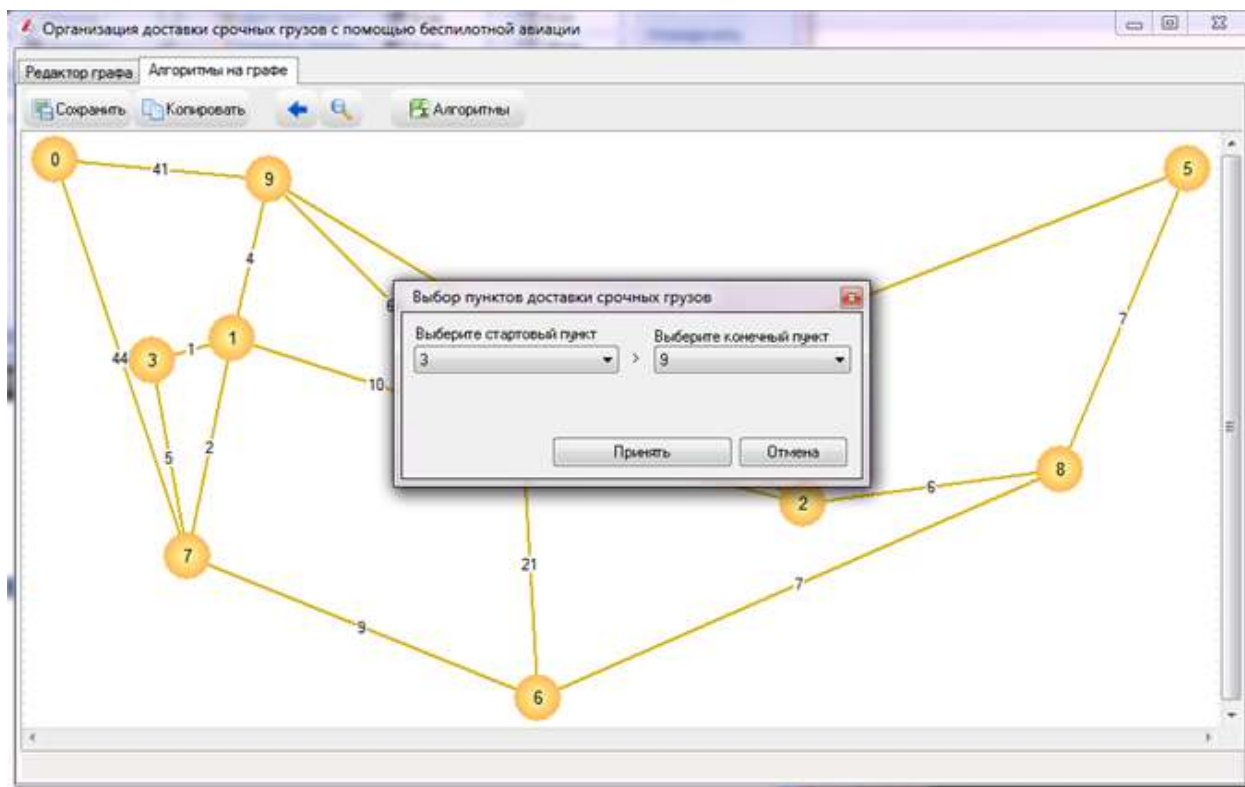


Рисунок 3.7 – Результат тестування. Вибір пунктів

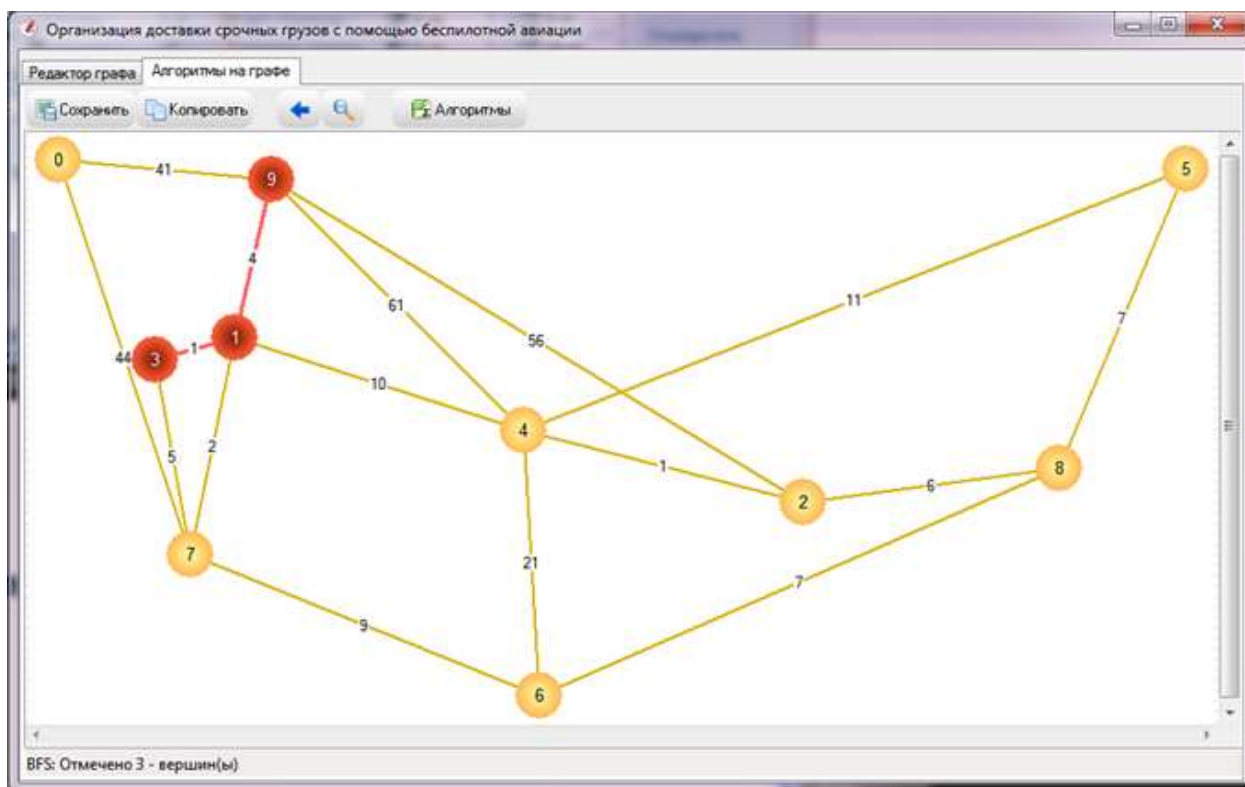


Рисунок 3.8 – Результат тестування

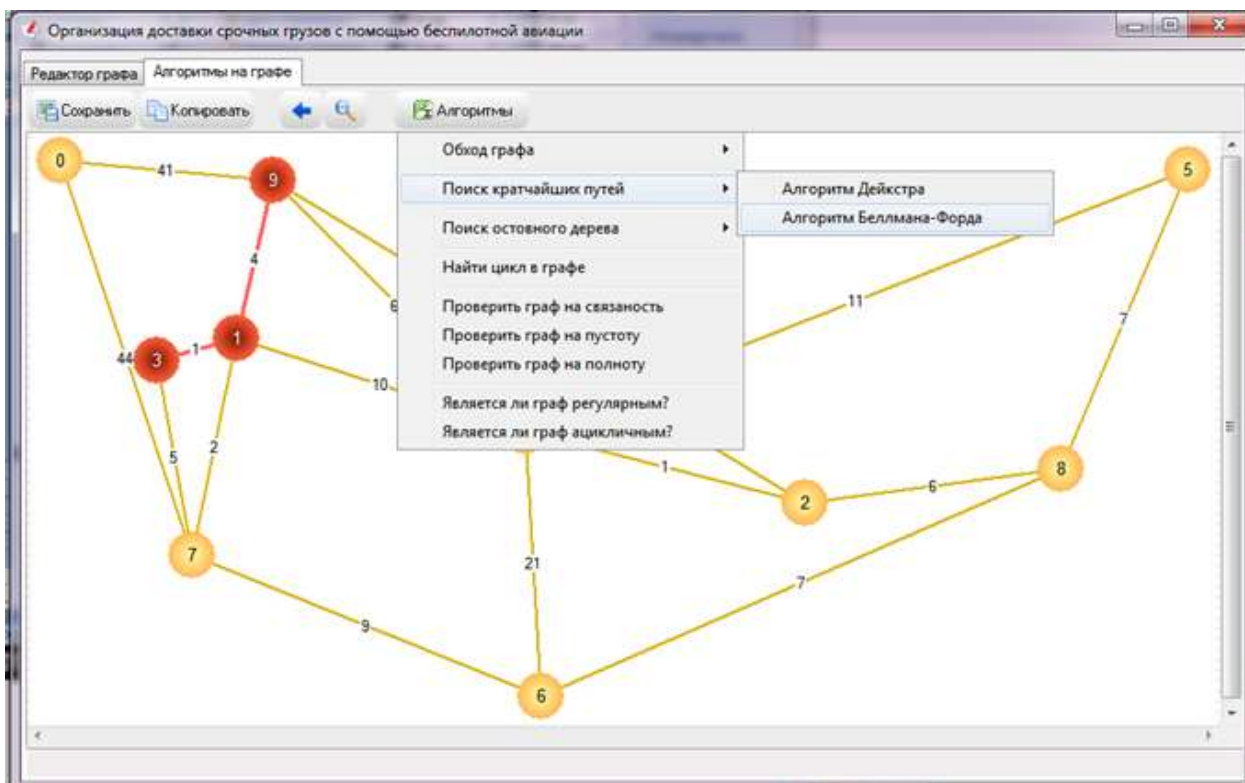


Рисунок 3.9 – Результат тестування

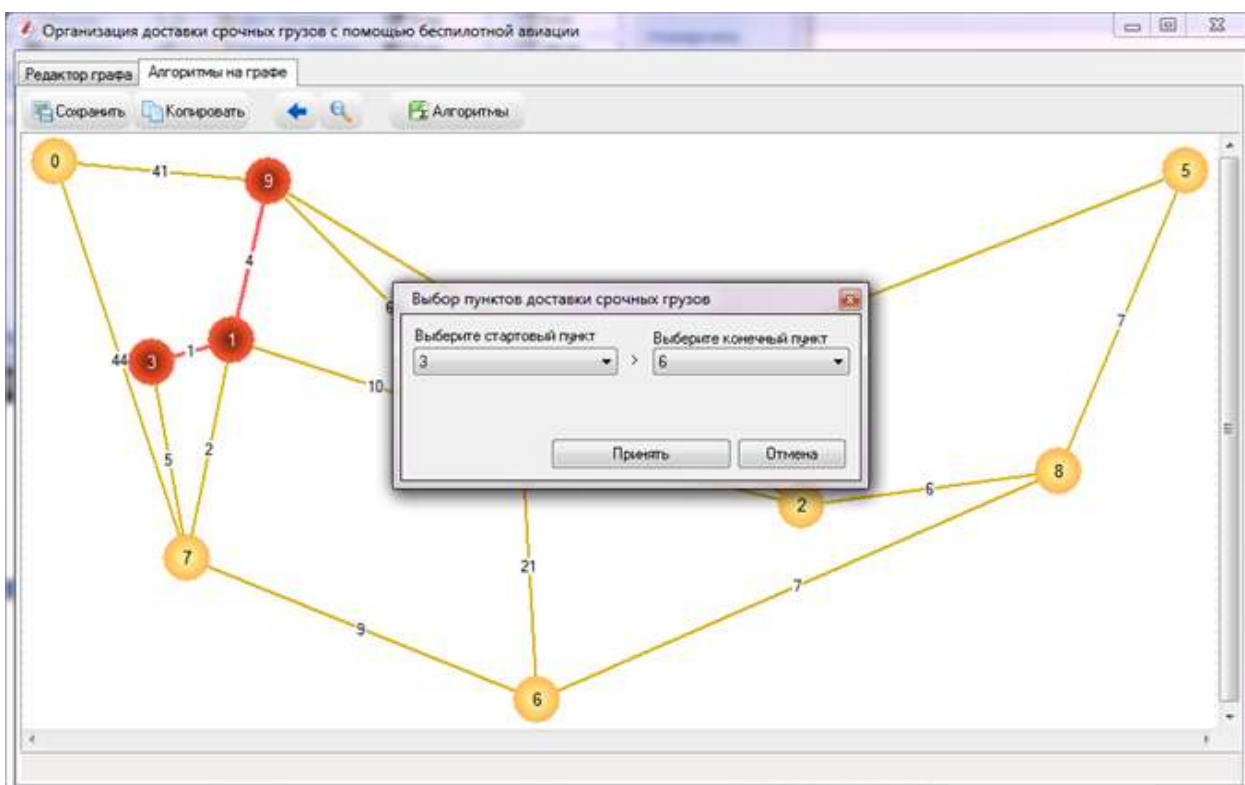


Рисунок 3.10 – Результат тестування

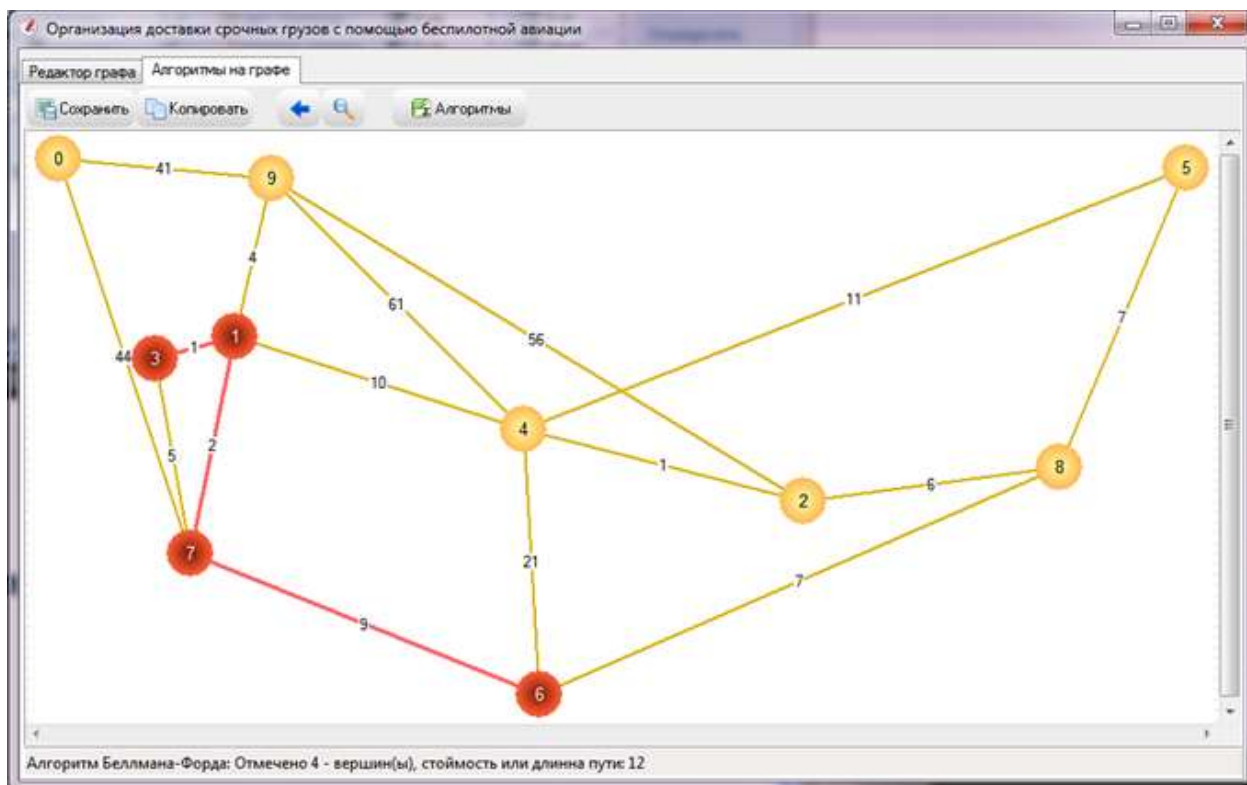


Рисунок 3.11 – Результат тестування

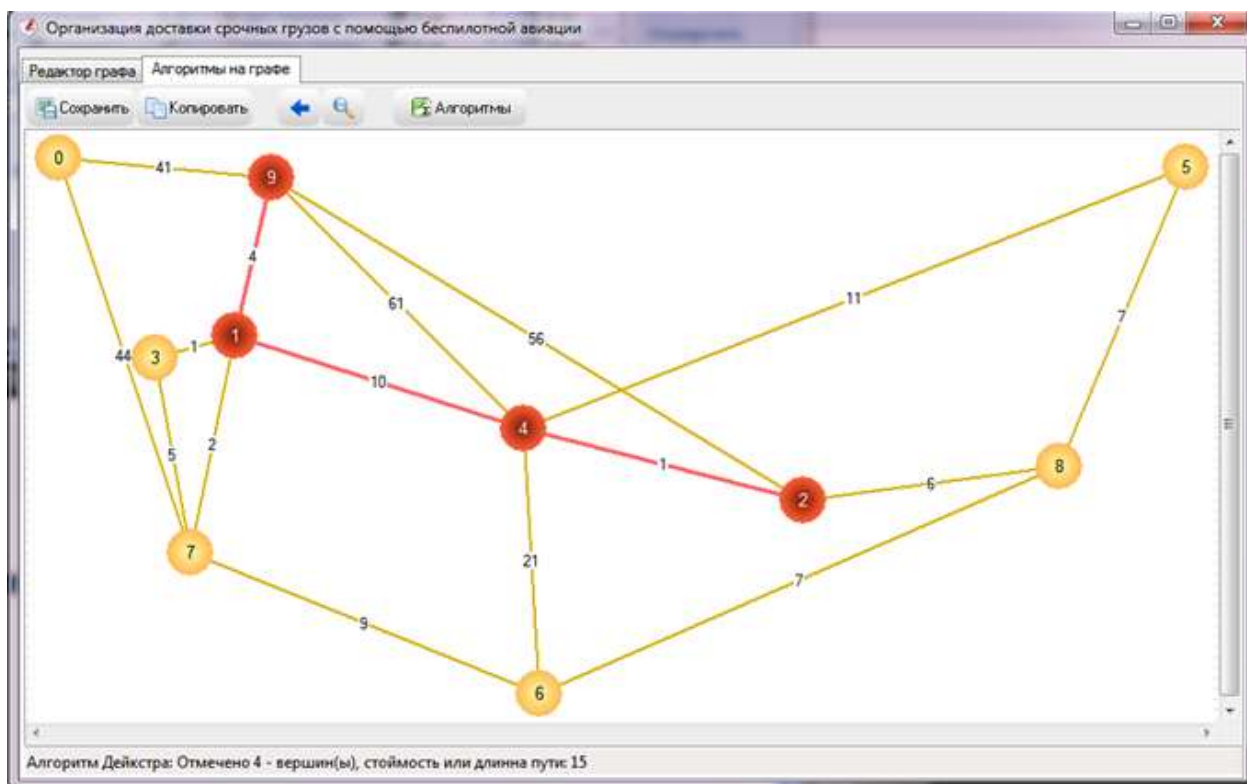


Рисунок 3.12 – Результат тестування

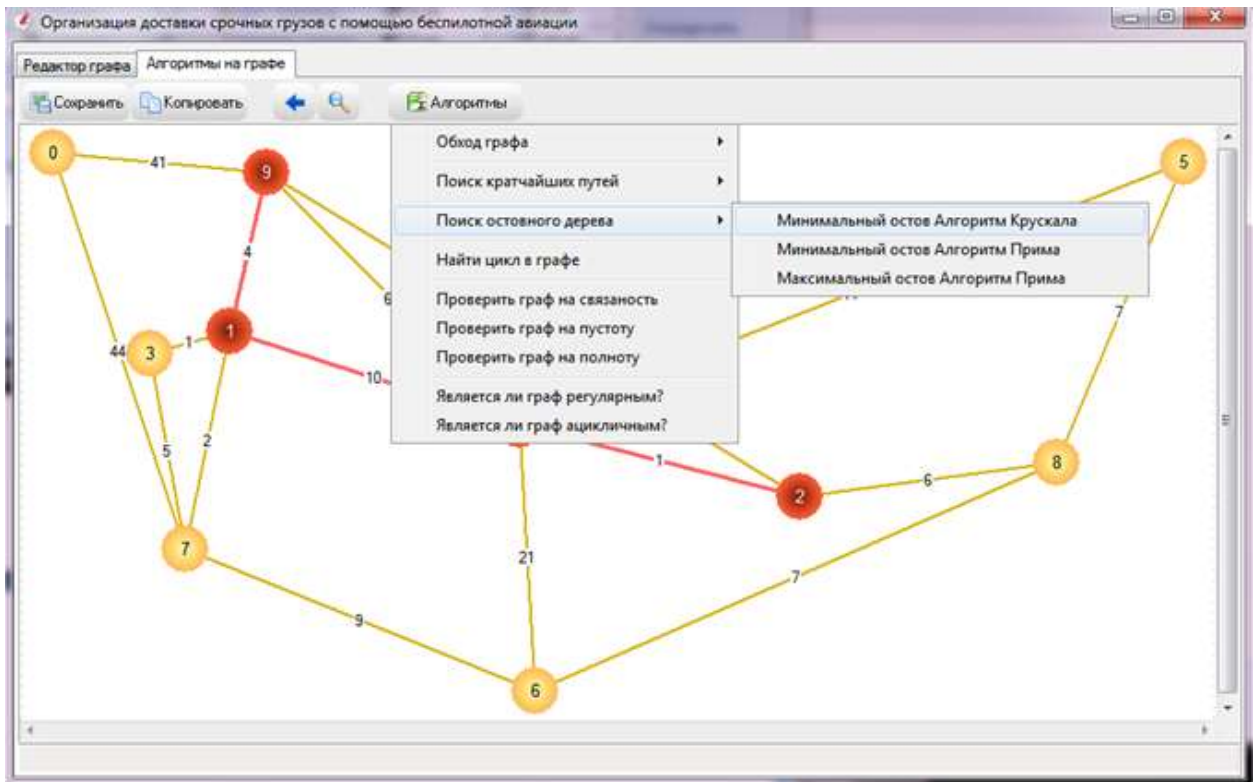


Рисунок 3.13 – Результат тестування

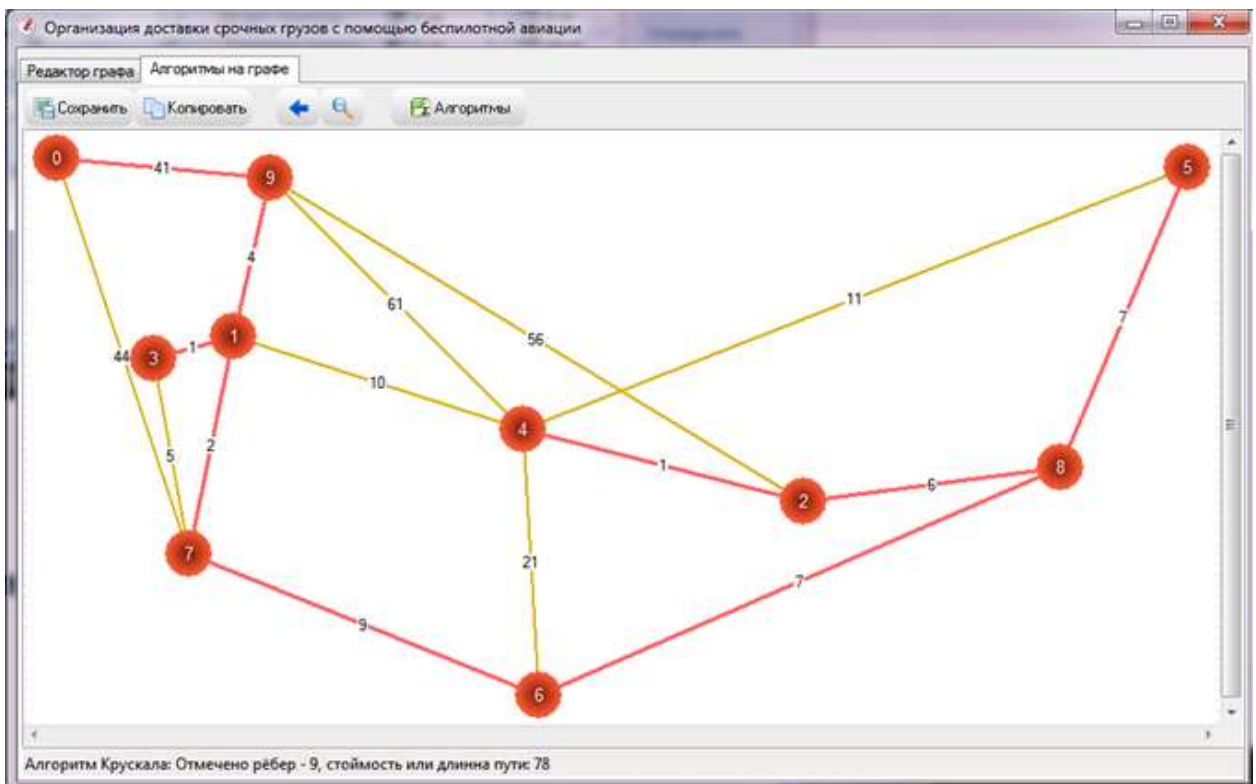


Рисунок 3.14 – Результат тестування

Під час тестування збоїв та недоліків у роботі системи організації доставки термінових вантажів за допомогою беспілотної авіації не виявлено,

що говорить про високу якість розробки та можливість впровадження за потребою.

### **Висновки до розділу**

У рамках третього розділу розкрито структуру організації доставки термінових вантажів за допомогою безпілотної авіації, описано основні модулі системи проведено тестування розробленої системи організації доставки термінових вантажів за допомогою безпілотної авіації. Під час тестування збоїв та недоліків у роботі системи організації доставки термінових вантажів за допомогою безпілотної авіації не виявлено, що говорить про високу якість розробки та можливість впровадження за потребою.

## *ВИСНОВКИ*

Кафедра ОАП				НАУ 20.02.20. 002 ПЗ			
Виконав	Петрук В.О.			ВИСНОВКИ	Літера	Аркуш	Аркушів
Керівник	Юн Г.М.				Д	110	1
Консульт.	Юн Г.М.				ФТМЛ 275 ОП-201М		
Н.Контр.	Жукова С.О.						
Зав. каф.	Юн Г.М.						

У межах даної роботи здійснено розробку системи організації доставки термінових вантажів за допомогою безпілотної авіації. На основі вищевикладеного варто зробити наступний висновок:

При розробці даного програмного продукту використовувалися наступні програмні засоби: C++; Блокнот. Використання розробленої моделі дасть можливість підвищити роботу менеджерів з перевезення термінових вантажів, а тим самим знизить кількість часу витраченого на складання маршруту та розрахунку витрат на перевезення термінових вантажів та підвищить якість роботи менеджерів з перевезення термінових вантажів на підприємстві.

Впровадження розробленого програмного додатку сприятиме зменшенню трудомісткості роботи менеджерів з перевезення термінових вантажів на підприємстві, автоматизації їх роботи при контролі та систематизації перевезень на підприємстві.

Під час тестування збоїв та недоліків у роботі системи організації доставки термінових вантажів за допомогою безпілотної авіації не виявлено, що говорить про високу якість розробки та можливість впровадження за потребою.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Полтавский А. В.* Методы, модели и системный синтез управления комплексами беспилотных летательных аппаратов двойного назначения: автореф. дис. ... д-ра техн. наук : 05.13.01 / А. В. Полтавский. – М., 2009. – 38 с
2. *Лучкин С. А.* Наземные пространственно-распределенные антенные системы радиолиний управления и передачи информации беспилотных авиационных комплексов [Текст]: автореф. дис. ... канд. техн. наук : 05.12.07 / С. А. Лучкин. – Казань, 2010. – 15 с.
3. *Летунов Д. А.* Оптимизация системы управления легкого беспилотного летательного аппарата по частотному критерию: автореф. дис. ... канд. техн. наук : 05.13.05 / Д. А. Летунов. – Рыбинск, 2011. – 16 с.
4. *Машнин М. Н.* Информационно-измерительная и управляющая система малоразмерного беспилотного летательного аппарата повышенной точности [Текст]: автореф. дис. ... канд. техн. наук : 05.11.16 / М. Н. Машнин. – Тула, 2013. – 20 с.
5. Современные радиотехнические средства борьбы с беспилотными летательными аппаратами в зоне проведения АТО // М. Н. Ясечко, А. В. Очкуренко, А. А. Ковальчук, Д. В. Максютя / Збірник наукових праць Харківського національного університету Повітряних Сил. – 2015. – № 3(44). – С. 54-57.
6. *Иванов И., Чадов И.* Содержание и роль радиоэлектронной борьбы в операциях XXI века // Зарубежное военное обозрение. 2011. – № 1. – С. 14-20.
7. *Балыко Ю.* Беспилотники НАТО в локальных конфликтах / Ю. Балыко // Военный парад. – 2008. – № 1. – С.38–39.
8. Методы формирования и фокусировки электромагнитного излучения для воздействия на радиоэлектронные средства / М.Н. Ясечко,



Дохов А.И., М.Г. Иванец, О.В. Тесленко / Под ред. М.Н. Ясечко. – Харьков: ХУПС, 2015. – 220 с.

9. *Dunn D. H. Drones: disembodied aerial warfare and the unarticulated threat / David Hastings Dunn // International Affairs. 2013. – Vol. 89. – P. 1237–1246.*

10. *Masood S. C.I.A. Leaves Base in Pakistan Used for Drone Strikes / Salman Masood // The New York Times : website. 2016. – Режим доступа. – <http://www.nytimes.com/2011/12/12/world/asia/cia-leaves-pakistan-base-used-for-drone-strikes.html> (Станом на 09.11.2016 року)*

11. *Веремеенко К. К., Головинский А. Н., Инсаров В. В. Управление и наведение беспилотных маневренных летательных аппаратов на основе современных информационных технологий / под ред. М. Н. Красильщикова, Г. Г. Себрякова. – М.: Физматлит, 2013. – 280 с.*

12. *Радиоэлектронная борьба/ А.И.Куприянов. – М.: Вузовская книга, 2013. – 360 с.:ил.*

13. *Mohinder S. Grewal, Angus P. Andrews. Kalman Filtering: Theory and Practice. John Wiley & Sons. Inc. 2001.*

14. *Родионов В. В. Помехоустойчивость РЛС в условиях радиоэлектронного подавления // Радиолокация, навигация, связь: докл. конф., Воронеж, 15-17 апреля 2014 г. Воронеж: НПФ «Саквояж», 2014. – Т. 3. – С. 1476-1485.*

15. *Филиппов А. А. Повышение эффективности управления беспилотного летательного аппарата в условиях помех / А. А. Филиппов, Д. А. Бажин, А. Н. Хлобыстов // Информационно-управляющие системы. – 2014. – № 6. – С. 45-50*

16. *Сизоненко И.М. О содержании компьютерных программ. Информатика и образование, 2008. – 355 с.*

17. *Иващенко Т.П. Система информационного обеспечения управления. – Мн.: Ураджай, 2009. – 536 с.*

18. Информационные системы в экономике: Учебник/ Под ред. проф. В.В. Дика. – М.: Финансы и статистика, 2006. – 245 с.
19. Информационные системы и технологии в экономике: Учебное пособие под ред. В.Н. Федорова, Кемерово Кузбассвузиздат, 2003 . – 436 с.
20. Карминский А.М., Нестеров П.В. Информатизация бизнеса. – М.: Финансы и статистика, 1997. – 416 с.
21. Кузнецов М., Симдянов И. С++ на примерах. – Спб.: «БХВ-Петербург», 2008. – С. 952.
22. Левин А., Самоучитель работы на компьютере. 10-е издание, Питер. – 2008. – 426 с.
23. Осмоловский В.В. Теория анализа хозяйственной деятельности. – Мн.: Новое знание, 2001. – 314 с.
24. Основы экономической информатики: Учебное пособие / А.Н. Морозевич, Н.Н. Говядинов, Б.А. Железко и др.; Под общ. Ред. А.Н. Морозевича. – Мн.: ООО «Мисанта», 2015. – 426 с.
25. Попов В.М. и др. Глобальный бизнес и информационные технологии. Современная практика и рекомендации / В.М. Попов, Р.А. Маршавин, С.И. Лапунов; Под ред. В.М. Попова. – М.: Финансы и статистика, 2015. – 272 с.
26. Программирование на С++ Win32: С. А. Любавин – Санкт-Петербург, НТ Пресс, 2008. – 576 с.
27. Разработка и оформление конструкторской документации РЭА. Справочное пособие / Романовского и др. – М.: 2004. – 246 с.
28. Степанова Е.Е., Хмелевская Н.В. Информационное обеспечение управленческой деятельности: Учебное пособие – М.: ФОРУМ: ИНФРА-М, 2002. – 425 с.
29. Ширяев Д., Аншелес В., Мочалин В. Сбор и обработка информации для принятия управленческих решений. «Открытые системы», 2001. – №4. – С. 34-40.

30. Экономическая информатика: Учебник/ Под ред. В.П. Косарева, Л.В. Еремина. – М.: Финансы и статистика, 2001. – 592 с.
31. *Архандельский А.Я.* Программирование в C++Builder 6. К: ЗОА Издательство БИНОМ, 2001. – 1151 с.
32. *Велюха А. М.* Проектування програмного забезпечення економічних інформаційних систем. – К.: Фінанси і статистика, 2012. – 634 с.
33. *Джесс Либерти* Освой самостоятельно C++. К: Издательский дом Вильямс, 2000. - 815 с.
34. *Журавкова І.В.* Аналіз ефективності діяльності підприємства. – К.: Фінанси і статистика, 2011. – 487 с.
35. *Крутик А.Б. Нікольська Є.Г.* Економіка підприємства – Львів: Лань, 2013. – 311 с.
36. *Лисовак Б.Г.* Розробка управлінського рішення. – К.: Справа, 2012. – 243 с.
37. *Негалова Є.В.* Методика фінансового аналізу. Х.: ДАНА, 2012. – 153 с.
38. C/C++. Программирование на языке высокого уровня / Павловская Т.А. – СПб.: Питер, 2008. – 464 с.
39. *Титоренко Г.А.* Інформаційні технології управління. – Х.: ДАНА, 2013. – 322 с.
40. *Уткін В.Б.* Інформатика: Підр. для вузів. – К.: Проект, 2012. – 304 с.
41. *Фіногеев А.Г.* Мережеві технології. Частина 1: Навчальний посібник. – Харків: Вид-во ПГУ, 2013. – 107 с.
42. *Ackerman W.B.* Data Flow Languages, Proceedings of the NCC, Montvale, NJ, AFIPS, 1979, p. 1087-1095.
43. *Arvind A.* Critique of Multiprocessing von Neumann Style. Proc. of 10th Annual Int. Symp. on Computer Architecture. 1983, p. 426-436.

44. *Dennis J. B., Misunas D. P. A Preliminary Architecture for Basic Data Flow Processor // Proc. of 2nd Annual Int. Symp. on Computer Architecture. – N.Y., 1975, P. 126-132.*

45. *Myers G.J. Advances in Computer Architecture. Second Edition. – JONH WILLEY & SONS. 1982. P. 98-111.*

46. Уроки Программирование на C++ с нуля. [Электронный ресурс].  
– Режим доступа: <http://code-live.ru/tag/cpp-manual/>

## ДОДАТКИ

### Додаток А

#### Лістинг програмного коду

```

#ifndef _KUDUSHTEEV_CONTROLS_H_
#define _KUDUSHTEEV_CONTROLS_H_
#if _MSC_VER > 1000
#pragma once
#endif
#define CLASSNAME_IMAGEBUTTON _T("xImageButton")

//контексное меню
class xPopupMenu {
private:
    HMENU menu;
public:

    xPopupMenu(void): menu(NULL){}
    ~xPopupMenu() {
        this->Destroy();
    }

public:

    void Create(void) {
        this->Destroy();
        menu = CreatePopupMenu();
    }

    void Add(const TCHAR* name, UINT id, bool mbreak = false) {
        AppendMenu(menu, (mbreak) ? (MF_STRING | MF_MENUBREAK) :
MF_STRING, id, name);
    }

    void Add(HMENU parent, const TCHAR* name, UINT id, bool mbreak =
false) {
        if((parent == NULL) || (! IsMenu(parent)))
            return;
        AppendMenu(parent, (mbreak) ? (MF_STRING | MF_MENUBREAK) :
MF_STRING, id, name);
    }

    HMENU AddSub(const TCHAR* name, UINT type = MF_STRING) {
        HMENU msub;
        if(menu == NULL)
            return NULL;

        msub = CreatePopupMenu();
        if(msub == NULL)

```

```

        return NULL;

        BOOL ret = AppendMenu(menu, type | MF_POPUP, (UINT_PTR)msub,
name);
        if(! ret) {
            DestroyMenu(msub);
            msub = NULL;
        }
        return msub;
    }

    void Show(HWND hwnd, int x, int y, UINT align = TPM_LEFTALIGN) {
        SetForegroundWindow(hwnd);
        TrackPopupMenu(menu, align, x, y, 0, hwnd, NULL);
        PostMessage(hwnd, WM_NULL, 0, 0);
    }

    void SetChecked(UINT id, bool enable) {
        CheckMenuItem(menu, id, MF_BYCOMMAND | ((enable) ?
MF_CHECKED : MF_UNCHECKED));
    }

    bool SetCheckRadio(UINT id_first, UINT id_last, UINT id) {
#ifdef WINVER >= 0x0400
        if(menu != NULL)
            CheckMenuRadioItem(menu, id_first, id_last, id,
MF_BYCOMMAND);
        return true;
#else
        return false;
#endif
    }

    void SetEnabled(UINT id, bool enable){
        UINT flag = (enable) ? MF_ENABLED : MF_DISABLED | MF_GRAYED;
        EnableMenuItem(menu, id, MF_BYCOMMAND | flag);
    }

    bool IsChecked(UINT id) {
        return (GetMenuState(menu, id, MF_BYCOMMAND) & MF_CHECKED) ?
true : false;
    }

    bool IsEnabled(UINT id) const {
        return (GetMenuState(menu, id, MF_BYCOMMAND) & MF_DISABLED)
? false : true;
    }

    void Destroy(void) {
        if(menu != NULL)
            DestroyMenu(menu);
        menu = NULL;
    }

};

```

```

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

```

```

class xToolTip {
private:
    HWND      c_wnd;
    TOOLINFO info;
public:

    xToolTip(void): c_wnd(NULL){}
    ~xToolTip() {
        this->Destroy();
    }

    xToolTip(const xToolTip&);
    xToolTip& operator = (const xToolTip&);
public:

    BOOL Create(HWND parent) {
        HINSTANCE inst = g_FromInstance(parent);

        c_wnd = CreateWindowEx(WS_EX_TOPMOST, TOOLTIPS_CLASS, NULL,
WS_POPUP | TTS_ALWAYSSTIP,
                                CW_USEDEFAULT,
CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, parent, NULL, inst, NULL);
        if(c_wnd == NULL)
            return FALSE;

        memset(&info, 0, sizeof(TOOLINFO));
        info.cbSize = sizeof(TOOLINFO);
        info.uFlags = TTF_IDISHWND | TTF_ABSOLUTE | TTF_TRACK;
        info.hwnd = parent;
        info.uId = (UINT)parent;
        info.hinst = g_FromInstance(c_wnd);
        info.lpszText = _T("xxxxxxxxxxxx");
        SendMessage(c_wnd, TTM_ADDTOOL, 0,
(LPPARAM) (LPTOOLINFO) &info);

        SendMessage(c_wnd, TTM_SETDELAYTIME, TTDT_RESHOW, 50);
        return TRUE;
    }

    void Show(TCHAR* str) {
        POINT pt;
        if(IsWindow(c_wnd)) {
            GetCursorPos(&pt);
            this->Show(pt.x, pt.y, str);
        }
    }

    void Show(int x, int y, TCHAR* str) {
        if(IsWindow(c_wnd)) {
            info.lpszText = str;

```

```

        SendMessage(c_wnd,          TTM_UPDATETIPTTEXT,      0,
(LPPARAM) (LPTOOLINFO) &info);
        SendMessage(c_wnd,          TTM_TRACKPOSITION,      0,
(LPPARAM) (LONG) MAKELONG(x, y));
        SendMessage(c_wnd,          TTM_TRACKACTIVATE,
(WPARAM) (BOOL) TRUE, (LPARAM) (LPTOOLINFO) &info);
    }
}

void Hide(void) {
    if(IsWindow(c_wnd))
        SendMessage(c_wnd,          TTM_TRACKACTIVATE,
(WPARAM) (BOOL) FALSE, (LPARAM) (LPTOOLINFO) &info);
}

HWND Handle(void) {
    return c_wnd;
}

void Destroy(void) {
    if(c_wnd != NULL)
        DestroyWindow(c_wnd);
    c_wnd = NULL;
}
};

```

```

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

```

```

class xComboBox {
    enum { COMBO_NONE, COMBO_RES, COMBO_NEW };
private:
    HWND c_wnd;
    int c_type;
public:

    xComboBox(void): c_wnd(NULL), c_type(COMBO_NONE) {}
    xComboBox(HWND hwnd, UINT id) {
        c_wnd = GetDlgItem(hwnd, id);
        c_type = COMBO_RES;
    }
    ~xComboBox() {
        this->Destroy();
    }

    xComboBox(const xComboBox&);
    xComboBox& operator = (const xComboBox&);
public:

    BOOL Create(HWND parent, UINT id, int x, int y, int width, int
height) {

```



```

        HINSTANCE inst = g_FromInstance(parent);

        c_wnd = CreateWindow(_T("COMBOBOX"), NULL, WS_CHILD |
CBS_AUTOHSCROLL | CBS_DROPDOWNLIST | WS_VISIBLE,
        x, y, width, height, parent, (HMENU)id,
inst, NULL);

        if(c_wnd == NULL)
            return FALSE;
        SendMessage(c_wnd, WM_SETFONT,
(WPARAM) GetStockObject(DEFAULT_GUI_FONT), MAKELPARAM(TRUE, 0));
        c_type = COMBO_NEW;
        return TRUE;
    }

    void FromHandle(HWND hwnd, UINT id) {
        c_wnd = GetDlgItem(hwnd, id);
        c_type = COMBO_RES;
    }

    void SetEnabled(bool enable) {
        if(IsWindow(c_wnd))
            EnableWindow(c_wnd, (enable) ? TRUE : FALSE);
    }

    //присвоить текст
    void SetText(const TCHAR* str) {
        SendMessage(c_wnd, WM_SETTEXT, 0, (LPARAM) (const
TCHAR*)str);
    }

    // ассоциировать дополнительные данные к указанному элементу
    void SetItemData(int index, DWORD data) {
        SendMessage(c_wnd, CB_SETITEMDATA, (LPARAM) index,
(LPARAM) (DWORD) data);
    }

    // добавить новую строку
    void Add(const TCHAR* str){
        SendMessage(c_wnd, CB_ADDSTRING, 0, (LPARAM) (const
TCHAR*)str);
    }

    // вставить новую строку на указанную позицию по индексу
    void Insert(int index, const TCHAR* str){
        SendMessage(c_wnd, CB_INSERTSTRING, (LPARAM) index,
(LPARAM) (const TCHAR*)str);
    }

    // удаление элемента по индексу
    void Delete(int index){
        SendMessage(c_wnd, CB_DELETESTRING, (LPARAM) index, 0);
    }

    // удаление всех элементов(очистка)
    void Clear(void) {
        SendMessage(c_wnd, CB_RESETCONTENT, 0, 0);
    }

```

```

}

// выбор указанного элемента по индексу
void SetSelectedIndex(int index){
    SendMessage(c_wnd, CB_SETCURSEL, (WPARAM)index, 0);
}

// установка лимита символов на строку
void SetLimitText(int maxlen) {
    SendMessage(c_wnd, CB_LIMITTEXT, (WPARAM)maxlen, 0);
}

//***** СВОЙСТВА ПОЛУЧЕНИЕ GET-ДАНЫХ *****
// поиск данных возвращая индекс элемента
int Find(const TCHAR* str, int pos = 0) {
    if(c_wnd == NULL)
        return -1;
    return (int)SendMessage(c_wnd, CB_FINDSTRING, (WPARAM)pos,
(LPARAM) (const TCHAR*) str);
}

// получение ассоциированных данных с указанного элемента по
индексу
DWORD GetItemData(int index) {
    if(c_wnd == NULL)
        return 0uL;
    return (DWORD)SendMessage(c_wnd, CB_GETITEMDATA,
(WPARAM)index, 0);
}

// получение элемента-строки по указанному индексу
void GetAt(int index, const TCHAR* buf){
    SendMessage(c_wnd, CB_GETLBTEXT, (WPARAM) index,
(LPARAM) (TCHAR*) buf);
}

// получение кол-во символов указанной строки-элемента
int GetLengthAt(int index){
    if(c_wnd == NULL)
        return 0;
    return (int)SendMessage(c_wnd, CB_GETLBTEXTLEN,
(WPARAM) index, 0);
}

// получение указанной выделенной строки-элемента
int GetSelectedIndex(void){
    if(c_wnd == NULL)
        return -1;
    return (int)SendMessage(c_wnd, CB_GETCURSEL, 0, 0);
}

// кол-во элементов
int GetCount(void) {
    if(c_wnd == NULL)

```

```

        return -1;
        return (int)SendMessage(c_wnd, CB_GETCOUNT, 0, 0);
    }

    //получить текст
    void GetText(TCHAR* dst, int maxlen) {
        SendMessage(c_wnd, WM_GETTEXT, (WPARAM)maxlen,
(LPARAM) (TCHAR*)dst);
    }

    //получение длины текста
    int GetLength(void) const {
        if(c_wnd == NULL)
            return 0;
        return GetWindowTextLength(c_wnd);
    }

    void Destroy(void) {
        if(c_type == COMBO_NEW) {
            if(c_wnd != NULL)
                DestroyWindow(c_wnd);
            c_wnd = NULL;
        }
        c_type = COMBO_NONE;
    }
};

////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////

// класс-объект "STATUSBAR"
class xStatusBar {
private:
    HWND c_status;
public:

    xStatusBar(void): c_status(NULL){}
    ~xStatusBar() {
        this->Destroy();
    }

public:

    // создание статус-бара(строки состояние)
    bool Create(HWND hwnd) {
        INT arr[1];
        RECT rc;
        c_status = CreateWindow(STATUSCLASSNAME, NULL, WS_CHILD |
WS_CLIPCHILDREN | WS_CLIPSIBLINGS,

```

```

                                0,    0,    0,    0,    hwnd,    NULL,
::g_FromInstance(hwnd), NULL);
    if(c_status == NULL)
        return false;
    ShowWindow(c_status, SW_SHOW);
    UpdateWindow(c_status);

    GetClientRect(c_status, &rc);
    arr[0] = rc.right;
    SendMessage(c_status, SB_SETPARTS, 1, (LPARAM)(LPINT)arr);

    GetClientRect(hwnd, &rc);
    SendMessage(c_status, WM_SIZE, 0,
(LPPARAM)(MAKELPARAM((WORD)rc.right, (WORD)rc.bottom)));
    return true;
}

// вывод текста на указанную панель строки состояние
void SetText(const TCHAR* str) {
    if(c_status != NULL)
        SendMessage(c_status, SB_SETTEXT, 0, (LPARAM)(const
TCHAR*)str);
}

//метод должен быть вызван в событие изменение размера формы
WM_SIZE
void OnSize(int width, int height) {
    INT arr[1];
    if(IsWindow(c_status)) {
        arr[0] = width;
        SendMessage(c_status, SB_SETPARTS, 1,
(LPPARAM)(LPINT)arr);
        SendMessage(c_status, WM_SIZE, 0,
(LPPARAM)(MAKELPARAM((WORD)width, (WORD)height)));
    }
}

HWND Handle(void) {
    return c_status;
}

//удаление всех ресурсов
void Destroy(void) {
    if(c_status != NULL) {
        DestroyWindow(c_status);
        c_status = NULL;
    }
}
};

```

```

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

```

```

////////////////////////////////////
////////////////////////////////////

```

```

//класс-объект "КНОПКА ИЗОБРАЖЕНИЕ"
class xImageButton {

    enum {
        STATE_DEFAULT = 0,
        STATE_ACTIVE = 1,
        STATE_PUSH = 2
    };

private:
    HWND    c_wnd;
    HWND    c_swnd;
    HDC     c_hdc;
    HBITMAP c_bmp;
    UINT    c_id;
    int     c_width;
    int     c_height;
    int     c_state;
    TCHAR*  c_hlp;

    xToolTip* ptool;
public:

    xImageButton(void) : c_wnd(NULL),
                        c_swnd(NULL),
                        c_hdc(NULL),
                        c_bmp(NULL),
                        c_id(0u),
                        c_width(0),
                        c_height(0),
                        c_state(STATE_DEFAULT),
                        c_hlp(NULL),
                        ptool(NULL) {}

    ~xImageButton() {
        this->Destroy();
    }

    xImageButton(const xImageButton&);
    xImageButton& operator = (const xImageButton&);
public:

    //создаём контрол
    BOOL Create(HWND parent, UINT id, int x, int y, int width, int
height, const TCHAR* hlp = NULL) {
        HDC     hdc;
        HRGN    rgn;
        RECT    rc;
        UINT    len;
        WNDCLASS cls;
        HINSTANCE inst;

        if(IsWindow(c_wnd))
            return false;
        memset(&cls, 0, sizeof(WNDCLASS));

        inst = g_FromInstance(parent);

```

```

if(! GetClassInfo(inst, CLASSNAME_IMAGEBUTTON, &cls)) {

    memset(&cls, 0, sizeof(WNDCLASS));
    cls.style          = 0;
    cls.lpfWndProc    = (WNDPROC)xImageButton::WndProc;
    cls.hInstance     = inst;
    cls.hCursor       = LoadCursor(NULL, IDC_ARROW);
    cls.lpszClassName = CLASSNAME_IMAGEBUTTON;
    cls.hbrBackground = NULL;

    if(! RegisterClass(&cls))
        return FALSE;
}

c_wnd = CreateWindow(CLASSNAME_IMAGEBUTTON, NULL, WS_CHILD
| WS_CLIPSIBLINGS | WS_CLIPCHILDREN,
                    x, y, width, height, parent, (HMENU)id,
inst, this);

if(c_wnd == NULL)
    return FALSE;
ShowWindow(c_wnd, SW_SHOW);
UpdateWindow(c_wnd);

rgn = CreateRoundRectRgn(0, 0, width, height, 14, 14);
SetWindowRgn(c_wnd, rgn, TRUE);
DeleteObject(rgn);

GetClientRect(c_wnd, &rc);
c_width = (int)rc.right;
c_height = (int)rc.bottom;

hdc = GetDC(c_wnd);
c_hdc = CreateCompatibleDC(hdc);
c_bmp = CreateCompatibleBitmap(hdc, c_width, c_height * 3);
SelectObject(c_hdc, c_bmp);
PatBlt(c_hdc, 0, 0, c_width, c_height * 3, BLACKNESS);
ReleaseDC(c_wnd, hdc);

c_hlp = NULL;
ptool = NULL;
if(hlp != NULL) {
    len = (UINT)_tcslen(hlp);
    c_hlp = (TCHAR*)HeapAlloc(GetProcessHeap(), 0uL,
(DWORD)(len + 1u) * sizeof(TCHAR));
    if(c_hlp != NULL) {
        _tcsncpy(c_hlp, hlp, len);
        c_hlp[len] = _T('\0');
    }
}
InvalidateRect(c_wnd, NULL, TRUE);
return TRUE;
}

BOOL Put(HINSTANCE hInstance, UINT id, const TCHAR* str) {
HICON icon = LoadIcon(hInstance, MAKEINTRESOURCE(id));
if(icon != NULL) {
    this->this_icon(icon, str);
    DestroyIcon(icon);
    InvalidateRect(c_wnd, NULL, TRUE);
    return TRUE;
}
}

```

```

        return FALSE;
    }

    BOOL Put(const TCHAR* filename, const TCHAR* str) {
        HICON icon = (HICON)LoadImage(NULL, filename, IMAGE_ICON,
0, 0, LR_LOADFROMFILE);
        if(icon != NULL) {
            this->this_icon(icon, str);
            DestroyIcon(icon);
            InvalidateRect(c_wnd, NULL, TRUE);
            return TRUE;
        }
        return FALSE;
    }

void SetMsg(HWND hwnd, UINT id, xToolTip* tip = NULL){
    c_swnd = hwnd;
    c_id = id;
    ptool = tip;
}

void Destroy(void) {
    if(c_bmp != NULL)
        DeleteObject(c_bmp);
    c_bmp = NULL;

    if(c_hdc != NULL)
        DeleteDC(c_hdc);
    c_hdc = NULL;

    if(c_wnd != NULL) {
        g_SetUserData(c_wnd, NULL);
        DestroyWindow(c_wnd);
    }
    c_wnd = NULL;

    if(c_hlp != NULL){
        HeapFree(GetProcessHeap(), 0uL, (LPVOID)c_hlp);
        c_hlp = NULL;
    }
}

private:

void this_icon(HICON icon, const TCHAR* str) {
    int len = (str == NULL) ? 0 : (int)_tcslen(str);
    HBRUSH hbr = this->get_bgcolor(c_hdc, RGB(0xFF, 0xFF,
0xFF), RGB(0xCC, 0xCC, 0xCC));
    this->this_button(0, str, len, hbr, icon, RGB(0, 0, 0));
    DeleteObject(hbr);

    hbr = this->get_bgcolor(c_hdc, RGB(0xFF, 0xFF, 0xCC),
RGB(0xDC, 0xCC, 0xEB));
    this->this_button(c_height, str, len, hbr, icon, RGB(0, 0,
0xFF), TRUE);
    DeleteObject(hbr);

    hbr = this->get_bgcolor(c_hdc, RGB(0xDC, 0xCC, 0xEB),
RGB(0xFF, 0xFF, 0xCC));

```

```

        this->this_button(c_height * 2, str, len, hbr, icon,
        RGB(0xAA, 0x24, 0), TRUE);
        DeleteObject(hbr);
    }

    void this_button(int top, const TCHAR* str, int len, HBRUSH hbr,
    HICON icon, COLORREF fcolor, BOOL bold = FALSE) {
        static const int fch = -MulDiv(8, GetDeviceCaps(c_hdc,
    LOGPIXELSY), 72);

        RECT rc;
        POINT pt;
        HFONT font;
        SIZE fsz;

        BOOL res = SetBrushOrgEx(c_hdc, 0, top, &pt);
        SetRect(&rc, 0, top, c_width, top + c_height);
        FillRect(c_hdc, &rc, hbr);

        int isz = 16;
        int ix = 0;
        if(len > 0) {

            font = CreateFont(fch, 0, 0, 0, FW_NORMAL, FALSE,
            bold, FALSE, RUSSIAN_CHARSET, OUT_DEFAULT_PRECIS,
            CLIP_DEFAULT_PRECIS,
            DEFAULT_QUALITY, DEFAULT_PITCH, _T("MS Sans Serif"));
            if(font == NULL)
                font = (HFONT)GetStockObject(DEFAULT_GUI_FONT);

            HGDIOBJ fold = SelectObject(c_hdc, font);
            int mode = SetBkMode(c_hdc, TRANSPARENT);

            if(! GetTextExtentPoint32(c_hdc, str, len, &fsz)){
                TEXTMETRIC tm;
                GetTextMetrics(c_hdc, &tm);
                fsz.cx = tm.tmAveCharWidth * len;
                fsz.cy = tm.tmHeight;
            }

            COLORREF color = SetTextColor(c_hdc, fcolor);
            ix = (c_width - (fsz.cx + isz)) / 2;
            TextOut(c_hdc, ix + isz + 2, top + (c_height -
            fsz.cy)/2, str, len);
            SetTextColor(c_hdc, color);
            SetBkMode(c_hdc, mode);
            SelectObject(c_hdc, fold);
            DeleteObject(font);
        } else
            ix = (c_width - isz)/2;

        if(res)
            SetBrushOrgEx(c_hdc, pt.x, pt.y, NULL);

        DrawIconEx(c_hdc, ix, top + (c_height - isz)/2, icon, isz,
        isz, 0, NULL, DI_NORMAL);
    }

    HBRUSH get_bgcolor(HDC hdc, COLORREF color1, COLORREF color2) {
        HDC mdc = CreateCompatibleDC(hdc);

```



```

c_height);
        HBITMAP  bmp    =    CreateCompatibleBitmap(hdc,    c_width,
        HBRUSH  hbr;

        SelectObject(mdc, bmp);

        const double step = 1.0 / (double)c_height;
        HGDIOBJ  old    =    SelectObject(mdc,
(HGDIOBJ)GetStockObject(DC_PEN));

        for(int i = 0; i < c_height; ++i) {
            SetDCPenColor(mdc,    Alg::ColorLerp(color1,    color2,
(double)i / c_height));
            MoveToEx(mdc, 0, i, NULL);
            LineTo(mdc, c_width, i);
        }
        SelectObject(mdc, old);

        hbr = CreatePatternBrush(bmp);
        DeleteObject(mdc);
        DeleteObject(bmp);
        return hbr;
    }

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

LRESULT ImageButtonProc(UINT msg, WPARAM wParam, LPARAM lParam) {
    RECT rc;
    POINT pt;
    TRACKMOUSEEVENT mev;

    switch(msg){
    case WM_LBUTTONDOWN:
        c_state = STATE_PUSH;
        InvalidateRect(c_wnd, NULL, TRUE);

        if(ptool != NULL)
            ptool->Hide();

        break;
    case WM_LBUTTONUP:
        c_state = STATE_ACTIVE;
        InvalidateRect(c_wnd, NULL, TRUE);
        SendMessage(c_swnd, WM_COMMAND, MAKEWPARAM(c_id, 0),
0);

        if(ptool != NULL)
            ptool->Hide();

        break;
    case WM_MOUSEMOVE:

        memset(&mev, 0, sizeof(TRACKMOUSEEVENT));
        mev.cbSize    = sizeof(TRACKMOUSEEVENT);
        mev.hwndTrack = c_wnd;
        mev.dwFlags    = TME_LEAVE;
        mev.dwHoverTime = HOVER_DEFAULT;
        TrackMouseEvent(&mev);
        if((c_state != STATE_ACTIVE)    &&    (c_state !=
STATE_PUSH)){

```

```

        c_state = STATE_ACTIVE;
        InvalidateRect(c_wnd, NULL, TRUE);
    }

    if((ptool != NULL) && (c_hlp != NULL)) {
        if(! IsWindowVisible(ptool->Handle())) {
            GetWindowRect(c_wnd, &rc);
            GetCursorPos(&pt);
            ptool->Show((int)pt.x, (int)rc.bottom +
10, c_hlp);
        }
    }

    break;
case WM_MOUSELEAVE:

    if(c_state != STATE_DEFAULT){
        c_state = STATE_DEFAULT;
        InvalidateRect(c_wnd, NULL, TRUE);
    }
    if(ptool != NULL)
        ptool->Hide();

    break;
case WM_ERASEBKGD:
    if(c_hdc != NULL)
        BitBlt((HDC)wParam, 0, 0, c_width, c_height,
c_hdc, 0, c_height * c_state, SRCCOPY);
    return 1;
case WM_DESTROY:
    this->Destroy();
    break;
default:
    return DefWindowProc(c_wnd, msg, wParam, lParam);
}
return 0;
}

static LRESULT CALLBACK WndProc(HWND hwnd, UINT msg, WPARAM
wParam, LPARAM lParam) {
    LPCREATESTRUCT ptr;
    xImageButton* obj;

    if(msg == WM_NCCREATE){
        ptr = (LPCREATESTRUCT)lParam;

        g_SetUserData(hwnd, ptr->lpCreateParams);
        ((xImageButton*)ptr->lpCreateParams)->c_wnd = hwnd;

        obj = (xImageButton*)ptr->lpCreateParams;
    } else
        obj = (xImageButton*)g_GetUserData(hwnd);

    if(obj != NULL)
        return obj->ImageButtonProc(msg, wParam, lParam);

    return DefWindowProc(hwnd, msg, wParam, lParam);
}
};

```

```

//класс-объект "СПИСОК КНОПОК ИЗОБРАЖЕНИЙ"
class xToolBar {

    struct node {
        xImageButton* ptr;
        node*         next;
    };

private:
    node*  head;
    node*  tail;

    void (*pfn_del)(xImageButton*);
public:

    xToolBar(void): head(NULL), tail(NULL), pfn_del(NULL) {}
    ~xToolBar() {
        this->Destroy();
    }

    xToolBar(const xToolBar& obj);
    xToolBar& operator = (const xToolBar&);

public:

    //добавить в голову списка за O(1)
    bool AddFront(xImageButton* ptr) {
        node*  pnew  = (node*)HeapAlloc(GetProcessHeap(), 0uL,
sizeof(node));
        if(pnew == NULL)
            return false;
        pnew->ptr  = ptr;
        pnew->next = NULL;
        if(head == NULL)
            head = tail = pnew;
        else {
            pnew->next = head;
            head = pnew;
        }
        return true;
    }

    //добавить в хвост списка за O(1)
    bool AddBack(xImageButton* ptr) {
        node*  pnew  = (node*)HeapAlloc(GetProcessHeap(), 0uL,
sizeof(node));
        if(pnew == NULL)
            return false;
        pnew->ptr  = ptr;
        pnew->next = NULL;
        if(head == NULL)
            head = tail = pnew;
        else {
            tail->next = pnew;
            tail = pnew;
        }
        return true;
    }
}

```

```

void SetFun(void (*pfn)(xImageButton*)) {
    pfn_del = pfn;
}

void Destroy(void) {
    node* tmp;
    while(head != NULL){
        tmp = head;
        head = head->next;

        if(pfn_del != NULL)
            (*pfn_del)(tmp->ptr);

        HeapFree(GetProcessHeap(), 0, (LPVOID)tmp);
    }
    tail = NULL;
}
};

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

// диалог
class xDialog {
private:
    HWND c_dlg;
public:

    xDialog(void):c_dlg(NULL){}
    ~xDialog() {}

    xDialog(const xDialog&);
    xDialog& operator = (const xDialog&);
public:

    int ShowModal(HINSTANCE hInstance, UINT id, HWND parent) {
        return DialogBoxParam(hInstance, MAKEINTRESOURCE(id),
parent, (DLGPROC)xDialog::DialogProc, (LPARAM)this);
    }

    void SetLocation(int x, int y) {
        if(IsWindow(c_dlg))
            SetWindowPos(c_dlg, HWND_NOTOPMOST, x, y, 0, 0,
SWP_NOSIZE | SWP_NOZORDER | SWP_NOACTIVATE);
    }

    void CloseResult(int ret) {

```

```

        if(IsWindow(c_dlg))
            EndDialog(c_dlg, ret);
    }

    void Close(void) {
        if(IsWindow(c_dlg))
            SendMessage(c_dlg, WM_CLOSE, 0, 0);
    }

    HWND Handle(void) {
        return c_dlg;
    }

    static LPVOID SetUser(HWND hwnd, LPVOID data){
#ifdef _WIN64
        return (LPVOID)SetWindowLongPtr(hwnd, DWLP_USER,
(LONG_PTR) data);
#else
        return (LPVOID)SetWindowLong(hwnd, DWL_USER, (LONG) data);
#endif
    }

    static LPVOID GetUser(HWND hwnd){
#ifdef _WIN64
        return (LPVOID)GetWindowLongPtr(hwnd, DWLP_USER);
#else
        return (LPVOID)GetWindowLong(hwnd, DWL_USER);
#endif
    }

protected:
    virtual void OnInitDialog(HWND hwnd) {}
    virtual void OnCommand(UINT id, UINT action) {}
    virtual void OnClose(void) {}
private:

    INT_PTR DlgProc(UINT msg, WPARAM wParam, LPARAM lParam) {
        switch(msg) {
            case WM_COMMAND:
                this->OnCommand((UINT) LOWORD(wParam),
(UINT) HIWORD(wParam));
                break;
            case WM_CLOSE:
                xDialog::SetUser(c_dlg, NULL);
                this->OnClose();
                EndDialog(c_dlg, 0);
                break;
        }
        return 0;
    }

    static INT_PTR CALLBACK DialogProc(HWND hwnd, UINT msg, WPARAM
wParam, LPARAM lParam) {
        xDialog* dlg;

        if(msg == WM_INITDIALOG) {
            xDialog::SetUser(hwnd, (LPVOID) lParam);

```

```
        dlg = (xDialog*)lParam;
        dlg->c_dlg = hwnd;
        dlg->OnInitDialog(hwnd);

        return 1;
    } else
        dlg = (xDialog*)xDialog::GetUser(hwnd);

    if(dlg != NULL)
        return dlg->DlgProc(msg, wParam, lParam);
    return 0;
}

};
#endif
```